Information Assurance in a Distributed Forensic Cluster

Processing Data for Forensics
in a
Distributed System

Nicholas Pringle MSc

A submission presented
in partial fulfilment of the requirements
of the University of South Wales/ /Prifysgol De Cymru
for the degree of Doctor of Philosophy

April 2015

Abstract

The first decade of the 21st Century has been described as a “Golden Age” in the development of Digital Forensics. Criminals naively used the new technology not realising they were leaving easy pickings for the investigators on their trail. The evidence was mostly obvious, the software straightforward. Most importantly, the scale of the task was manageable. A “Case” was more often, one suspect, one investigator, one computer, one hard disk, one piece of analysis software, one report for one Authority. The Golden Age is over. Investigations are becoming increasingly multi-jurisdictional, with multiple items containing evidence from multiple suspects and ever increasing quantities of data. Investigators are struggling to keep pace with the changes and are possibly losing the battle.

There have been several solutions proposed to regain the upper hand, amongst them is using what is collectively known as distributed processing running on clusters of PCs. Processing data in a forensically sound manner acceptable to the courts requires special measures when handling evidence. Existing systems in this area, like Hadoop and HTCondor, are designed for use in cases where users do not have to justify their actions to a legal authority. Appropriate procedures to attain a suitable “Chain of Evidence” have been developed as new forms of digital evidence have been identified, acquired, processed and presented in court. In these, the computer system used for analysis has been treated as a single point in the chain of evidence but in a distributed system, there could be hundreds of hosts connected via local and wide area networks. Currently no acceptable methods can assure “Chain of Evidence” in these ‘new’ distributed architectures.

Within this thesis, we present a solution to this problem. FCluster and FClusterfs are the result of a design research methodology that addresses the problem by setting a design criterion, proposing a design, building it and then evaluating it against a number of metrics identified in the background.

We find that, to be a complete solution, FCluster has to extend from the Acquisition of evidence through ingestion, distribution to processing. To overcome the latency problems common to distributed system we introduce a technique we call Jigsaw imaging and with it the prioritisation of data acquisition. It is implemented as a middleware, in a manner similar to Hadoop and HTCondor.

This dissertation makes an original contribution to knowledge in the field of digital forensics by developing a technique that ensures the integrity of data as it passes from acquisition source, to storage and on to processing within a distributed computer architecture.

Acknowledgments

I must first thank my wife, Val, who has put up with me over the last 3 years. The path of this research has not been straightforward or easy. Without her understanding, it would not have been completed.

I want to thank Miki, my supervisor, who, in truth didn't initially understand what I was doing, just gave me the space to get on with it and gave wise advice when she could. Latterly, when she did understand what I was doing, gave me even better advice.

I also wish to thank the European Union for giving me the money and KESS for their administration during the last 3 years, without which this would not have happened.

ESF

Llywodraeth Cymru
Welsh Government

Ewrop & Chymru: Buddsoddi yn eich dyfodol
Cronfa Gymdeithasol Ewrop

Europe & Wales: Investing in your future
European Social Fund

kess

Ysgoloriaethau Sgiliau Economi Gwybodaeth
Knowledge Economy Skills Scholarships

Contents

List of Figures

List of Tables

Glossary, Definition of Terms and Acronyms

Analogue Forensics	A term that describes all forensics that is not digital. This would include subjects such as ballistics, explosives, toxicology, mark, fingerprints, hand writing analysis and any chemical analysis that takes place in a laboratory,
3V	Volume, Velocity and Variety
ACPO	Association of Chief Police Officers
AFF	Advanced Forensic Format
AIP	Archival Information Packages
BLOB	Binary Large Object
BOINC	Berkeley Open Infrastructure for Network Computing
CIA	Confidentiality, Integrity, Availability
CIFS	Common Internet File System
CUDA	Compute Unified Device Architecture
DCO	Device Configuration Overlay
DEB	Digital Evidence Bag. Used interchangeably with DEC
DEC	Digital Evidence Container. Used interchangeably with DEB
DEFR	Digital Evidence First Responder. A specialist trained in the identification, securing and collection of digital evidence
DELV	Digital Environment for Large Scale Investigation
DES	Digital Evidence Specialist. An analyst who, under the instruction of the Sleuth, extracts information from the digital evidence collected by the DEFR
DFRWS	Digital Forensics Research Workshop (Conference)
Digital Forensics	A term used to differentiate digital forensics from its long-standing parent analogue forensics. This includes analysis of static media, dynamic memory, network traffic also analysis of data stored in a digital form for example digital images, email, database or financial records.
DIP	Dissemination Information Package
DVD	Digital Versatile Disc
ERR	Evidence Relevance Rating
FADM	Forensic Discovery Auditing Module
FAT	File Allocation Table
FSR	The Forensic Science Regulator
FTP	File Transfer Protocol
FUSE	File System in User Space
GB	Gigabyte
GDF	Generic Distributed Framework
GENA	General Event Notification Architecture
GPU	Graphical Processor Unit
HDFS	Hadoop Distributed File System
HPA	Host Protected Area
HPC	High Performance Computing
HRA	The Human Rights Act 1988
HTCU	High Tech Crime Unit
Imaging	The process of making a bitwise copy of storage media
ISO 14721:2012	An ISO standard that defines the Space Data and Information Transfer Systems - Open Archival Information System (OAIS) - Reference Model
ISO 17020:2012	An ISO standard that defines the Requirements for the operation of various types of bodies performing inspection
ISO 17025:2005	An ISO standard that defines the General requirements for the competence of testing and calibration laboratories
ISO 27001:2013	An ISO standard that defines an Information Security management system
ISO 27037:2012	An ISO standard that defines the Guidelines for identification, collection, acquisition, and preservation of digital evidence

KFF	Known File Format
LBA	Logical Block Address
LEF	Logical Evidence File
LOTA	Latency-Optimised Target Acquisition
MB	Megabyte
MBR	Master Boot Record
METS	Metadata Encoding and Transmission Standard
MFT	Master File Table
MDI	Multiple Document Interface
MIMD	Multiple Instruction, Multiple Data
MPI	Message Passing Interface
NCA	National Crime Agency
NFS	Network File System
NIST	National Institute of Standards and Technology
NTFS	New Technology File System
OAIS	Open Archival Information System
ONS	Office of National Statistics
PDI	Preservation Description Information
ROfs	Read Only file system
SIP	Submission Information Package
Sleuth	The investigator, often a Police Officer who may know relatively little about computers but whose strengths lie in their ability to solve a case
SMB	Server Message Block
SMP	Symmetric Multicore Processing
SOAP	Simple Object Access Protocol
SoC	System On a Chip
SSD	Solid State Devices/Drives
SSH	Secure Shell
SSL	Secure Sockets Layer
TB	Terabyte
TLS	Transport Layer Security

1 Introduction

1.1 Introduction

The scale of the creation, capture, processing, storage and availability of digital data is now beyond all but the wildest predictions of only a working lifetime ago. This has bought great changes in our lives. We have huge volumes of knowledge readily available. Media and entertainment has been transformed. Communication on a global scale is now as easily as communication with our geographic neighbours. There are 'gadgets' which were only previously found in the realm of science fiction. On the near horizon are intelligent machines that will work autonomously of human control.

Unfortunately, this has not come without a downside. Inevitably, criminal endeavour has followed technological developments to establish a new domain - cybercrime. Digital information is just a tool and can be used for good or bad. Old crimes, like hate mail, now have a digital form and new digital crimes, like hacking, now exist.

In response to these new crimes, new methods of detection and investigation have been developed to identify, gather evidence and prosecute these criminals. This is no small task and investigators are becoming overwhelmed by the sheer quantity of data to examine.

In this dissertation, we present a review of the current literature in the area of processing data for forensics. We then argue that the need to process this increase in data volume will lead to a move towards the adoption of distributed processing and cloud computing as an investigative tool but this will lead to a loss of assurance, specifically Chain of Evidence, and so result in the inadmissibility of evidence in court.

We therefore present a distributed middleware system, FCluster, which incorporates methods that provide assurance when handling data in these new architectures. These assurance methods are similar to exiting methods used in current analogue and digital forensics and so should not be too unfamiliar to what is a deeply conservative domain.

1.2 Background

1.2.1 The Start of Digital Forensics

Digital crime and digital forensics is young. The first law specifically crafted for digital crime was the Florida Computer Crime Act in 1978 (HistoryofInformation.com 2014). The FBI created its Computer Analysis and Response Team 6 years later in 1984 (FBI 2014a). In 1999, the Association of Chief Police Officers published the first Good Practice Guide for electronic evidence.

During this period, most of the tools used in an investigation would have been derived from the software used by software developers or support staff. Norton Disk Editor was, for a long time, a standard item in the arsenal of the digital investigator. In the mid-1990s and early 2000s software specifically designed for digital investigation was developed, like EnCase from Guidance Software (Guidance Software 2014) and FTK from AccessData (2014). In the middle of the first decade of the 21th century, it looked as if equilibrium had been established. Broadly, the tools suited the task. Then, it started to change.

1.2.2 The Information Age

At the turn of the millennium, probably as the result of the availability of the Internet, what had been predicted as the Information Age started to become apparent. It is difficult to quantify the information explosion that has taken place and is still in progress. In 1980, a business might own one Apple II that had one floppy disk drive. In 2013, an ‘ordinary person’ quite possibly owns a desktop computer that may well have a one Terabyte hard disk drive. That is an increase of 6 million fold in about 35 years. Many people own multiple devices; a Smartphone, a Desktop PC, a Tablet, an MP3 music player, a GPS and multiple USB memory sticks and memory cards for cameras and videos. This is not limited to people in the ‘wealthy 1st World’. People in the emerging markets are not going to wait 30 years to get their digital devices.

When this research started, in 2007, manufacturers were offering 500GB drives for £99 and were starting to offer terabyte storage. In 2013, it became difficult to buy a drive under 500GB and in 2014 a six Terabyte drive retails for £125. While the capacity of the media has increased by about 5 million fold, the I/O rate has increased from 5 MB/s with MFM to 600 MB/s with SATA-III, an increase of only 160 fold (Wikipedia 2013).

The result of these changes is that accessing all the data on the media, a process almost unique to digital forensics, is taking very much longer.

1.2.3 Recent cases

During the summer of 2006, when there was the potential for a terrorist bombing campaign against transatlantic flights, the security clampdown caused huge confusion for travellers and financial loss for the travel industry. By way of explanation for their action, during a televised press conference (BBC 2006), DCC Peter Clarke stated that the Metropolitan Police had made 27 arrests and subsequently seized 400 personal computers, 200 mobile phones and 8,500 items of digital evidence (presumably CDs, DVDs, memory sticks etc.). The statement went further to estimate this represented 6TB of data. This was huge by the standards of the day. If this estimate was true, the devices must have been several years old and most of the extra media must have been floppy disks or perhaps CDs. In a verbal statement, broadcast on BBC radio on 15th June 2007 after the successful conviction of the terrorists, Peter Clarke praised the efforts of the digital investigative team saying that some officers, drawn in from the whole of the UK, spent the nights during the investigation in sleeping bags on the office floor. Such was the pressure of work to complete a substantial amount of the investigation with 14 days before the suspects had to be charged or released. If the 2007 investigation happened today, it could be 600TB of media.

In 2014, the BBC (BBC 2014b) reported on a Freedom of Information request made by the National Society for the Prevention of Cruelty to Children that asked every Police Constabulary in the UK about the volume of data seized in connection with child abuse investigations. Taken as a national figure, this is approaching 40,000 hard disks each year.

Figure 1 - Subject Domain Timeline is a compilation of landmarks, often drawn from memory, which shows the recent history of relevant items.

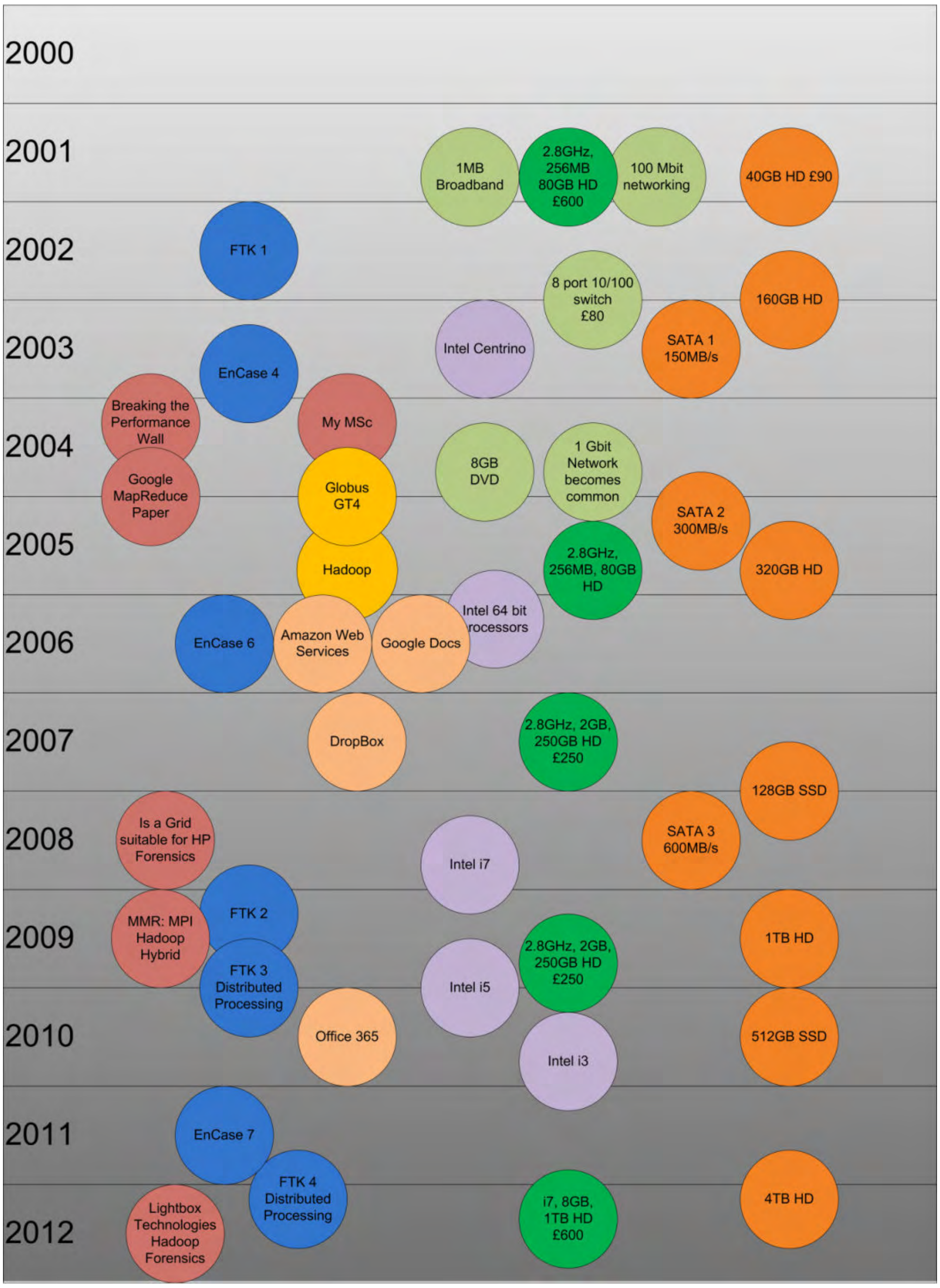


Figure 1 - Subject Domain Timeline

1.2.4 Growing Timescales from Data to Information to Knowledge

It is currently expected, as best practice from the Association of Chief Police Officers (ACPO) Guidelines for Digital Evidence (ACPO 2014), that investigators should start by imaging the entire storage media and then, as a subsequent task, run some analysis/indexing programs on the copy. Ultimately, it falls to an experienced human investigator to analyse the data, extract evidence, and form a case. From our own experience, in 2013, imaging a 4-terabyte hard disk drive takes about 17 hours and subsequent processing of the data from within the image to generate abstracted information takes about 4 days depending on the content of the media. AccessData's FTK can process about 2MB/s on a single core, so an i7 processor can process about 14 MB/s and a high-end Xeon processer with 32 cores can process about 64 MB/s. How long it takes the investigator to analyse that information to gain knowledge about a case is an open-ended question. Ironically, the most powerful computer become, it seems the longer it will take us to analyse 'the whole of the media'.

Analysis falls somewhere between two extremes. Sometimes the investigator is asked to locate very specific information, for example an email sent at a specific time. In this case, they can go straight to the email file and locate the evidence. At the other extreme, the investigator is asked to provide all the data relating to 'drugs'. This requires exhaustive sorting, selecting and indexing. It is a more complex search for interrelated information of evidential value. With four terabytes of assorted data, this is a formidable task. In these circumstances, pre-processing is essential but this could take weeks.

Current processing is quite basic. At most, current software attempts to recover lost or deleted file and then together with the 'normal' data from the file-system, organise these files into some classification to ease the manual search that follows.

More advanced processing is starting to appear, AccessData has just introduced an add-on to FTK that uses artificial intelligence to assess images for pornographic content but this is in its early days.

Perhaps there is searching with regular expressions and some search software allows indexing and fuzzy logic and stemming. Files can be compared using their cryptographic hashes but software that analyses images and recognises faces or places in some kind of automatic intelligent way, as humans do so easily, is currently only in the development stage. There is still sometime before we see the automated semantic machine understanding of emails or documents

During the last decade, there has been a significant shift in our aspirations for data processing. Whereas one of the historic skills of data processing was the discerning selection of source data and the efficient deployment of computing recourses to match the data processing load, there now seems to be a desire to adopt a 'catch it all' approach and with the idea that it can be just thrown into an expandable, dynamic processing solution. Digital forensic analysis is has not been spared from this thinking. Investigators are often expected to satisfy the desire to 'process everything'.

1.2.5 Possible Solutions

A number of solutions have been proposed. It would seem obvious to use computers that are more powerful but increasing a 'single box' solution to multi processors, large RAM and fast RAID type disk arrays does not achieve much more than a 10-fold increase in power but may result in a disproportionate 1,000 fold increase in price. Data triage, data reduction and machine intelligence are also key areas.

Digital forensics is not the only domain with this type of problem. The emerging domain of Big Data has the same types of problems and seems to have solved them by applying Distributed Processing techniques. Using Distributed Processing has been suggested as a solution in digital forensics but, as yet, no solutions has addressed all the issues that would make the solution acceptable in legal proceedings. AccessData's FTK4 system does offer what they call 'Distributed processing'. However, this is based on a central file server containing an image of the media and so is not truly distributed. This approach is not scalable to more than about ten processing hosts.

What is needed is a distributed platform upon which it is possible to run analysis software in a forensically sound environment. New tools need to have large amounts of processing power available to them. The platform needs to be able to make processing power available of a scale many hundreds, if not, thousands of times greater than the processing power used by the suspects.

1.2.6 Information Assurance

Users of existing popular distributed systems understand that they need to enforce their own standards and methods of assurance in their processing. The user has to check that they are using the correct data, that they are using the correct software and that this software is appropriate and up to date. Existing distributed systems are generally designed to be flexible and open for the user.

The legal process sets itself standards of assurance higher than those in most other domains. Procedures to ensure adequate assurance for Chain of Evidence in analogue forensics have been established to everyone's satisfaction. This model has been extended to digital forensics, but the assumption is that the investigator's computer is singular, at most a small network of PCs. Apart from possibly forbidding it, current 'Best Practice' in chain of evidence does not address the issues arising from the dispersal of evidence across a wide area network.

1.3 Research Method, Goals and Contributions

1.3.1 Motivations for Research

When society or the State, accuses, tries, convicts and enforces a penalty, which could be loss of liberty or even death in some jurisdictions, the accused has a fundamental human right to expect the evidence to be gathered, processed, presented and assessed with the very greatest of care possible. There is a great danger that in the enthusiasm to use the apparently impressive evidence revealed by digital forensic science, standards could be lowered in its processing. It does not take too much research to uncover important injustices that have their roots in inadequate assurance when processing evidence. This research strives to provide a universal base in a processing system that, at least, starts to build an assurance system that does not rely on individual diligence to maintain standards.

1.3.2 Problem Summary

- Various authorities (Parsonage 2009) (Garfinkel 2010) have reported a backlog of data requiring forensic analysis;

- This backlog is a problem as it delays prosecution and most legal frameworks have restrictions on the time suspects can be held without charge. Additionally, the value of intelligence information gained in an investigation usually degrades with time;

- This backlog is often caused by sheer quantity of data;

- Currently automation is very basic and most analysis is still manual. Sophisticated automation will require greater processing power and further extend processing times;

- This is largely because we are still using computer architectures that are limited by a disparity between disk capacities and interface transfer speeds and processing and data bus speeds;

- Many observers expect the backlog to increase;

- We believe fully distributed storage and processing is the only sustainable solution;

- Current Distributed Storage/Processing platforms do not provide adequate assurance for the legal process.

1.3.3 Scope

This research addresses the need for an assured process to control and monitor the introduction, passage, continued integrity and availability for processing of data introduced into a distributed cluster with the understanding that the results would be used in legal proceedings.

- We will assess the current process for acquiring data and introducing it to a computer system;
- We address the issues of a majority of computer devices that Law Enforcement meets regularly. Typically, these are devices running Microsoft Windows that use NTFS file system, Apple IOS and OSx that use HDFS, Linux that uses EXT3 and EXT4;
- We address the growing issues that come with large media over, typically 2TBytes.

This project focuses on the central, Middleware layer, as shown in Figure 2.

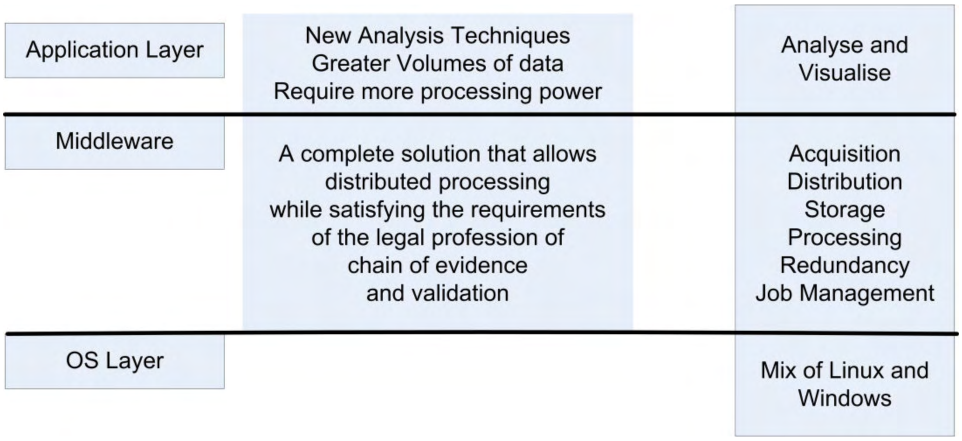


Figure 2 - System layers

1.3.4 Research Hypothesis

It is possible to facilitate the timely handling of large-scale digital evidence for professional computer forensic investigations, whilst still maintaining an appropriate chain of evidence, through the design of a suitable acquisition and processing methodology, implemented within a distributed middleware architecture.

1.3.5 Aims and objectives

Although there are examples of very highly funded organisations that are capable of building huge distributed systems of hundreds of thousands of hosts, this is not representative of the bulk of forensic workshops used by local or regional law enforcement. We choose to describe our target system by expressing it in terms of a budget that in 2014 might be £30,000. This could currently buy a cluster of about 100, i7 machines with 16GB RAM and 2TB of hard disk space. They would be connected with Gigabit Ethernet networking via a switch.

This research aims

1. To derive a set of requirements to enable the development of a distributed management system specifically suited to forensic investigation.
   This is evaluated in 8.7.1;
2. To evaluate some prominent existing distributed management systems and assess their suitability to implement a prototype distributed forensic system.
   This is evaluated in 8.7.2;
3. To classify existing forensic investigation tools and assess the likelihood of running them in a distributed environment and if necessary, to derive a standard for new tools intended to run within a distributed environment.
   This is evaluated in 8.7.3;
4. To develop a robust design of a middleware framework to support processing digital forensic tools in a distributed environment.
   This is evaluated in 8.7.4;
5. To evaluate the prototype system using representative case data.
   This is evaluated in 8.7.5.

It should be emphasised that the project is intend to develop a platform for upon which forensic tools can run, not the tools themselves, although we do intended to utilise some simple forensic Linux utilities to demonstrate the viability of the architecture.

1.3.6 Contributions to Research

The move to distributed processing for forensic analysis has not been as rapid as one might have expected a decade ago. It seems that a potential blockage in its development and adoption, is that the distributed storage of data, required to enjoy the full advantages of the distributed processing, has not been implemented to the satisfaction of the stakeholders who would be expected to challenge any change of processing paradigm. This research offers a release to this blockage by specifically addressing assurance in a distributed storage system.

1.3.7 Research Methodology

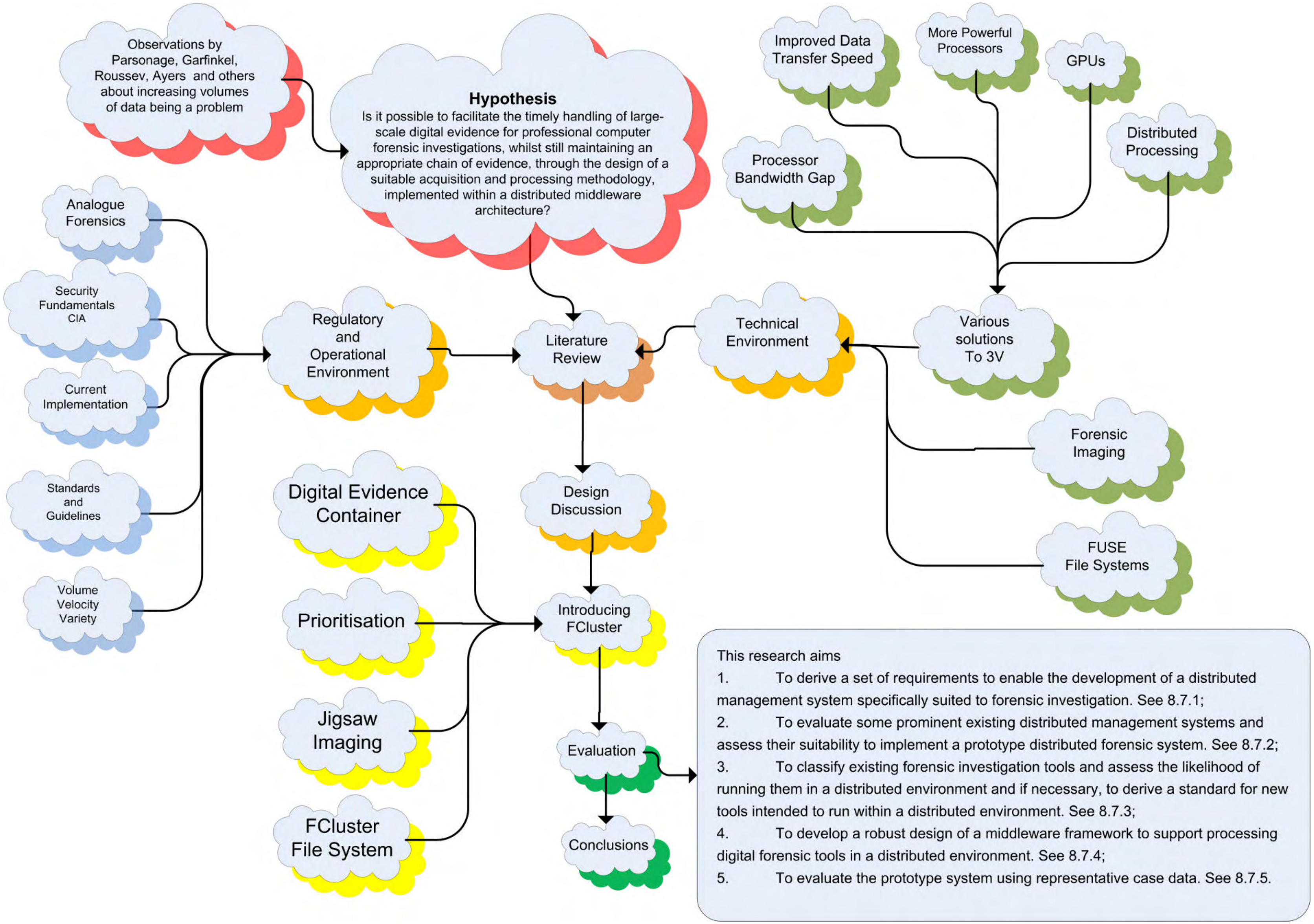
We have chosen to use ‘Design Research’ as a methodology for this project. It is frequently used within engineering and has been growing in popularity in the last few decades because of its alignment with a desire to produce a result that can be applied to real world problems. The methodology and its implementation are described in detail in chapter 2.

1.3.8 Deliverables

We will produce this dissertation and an implementation of the system as a VMWare image based on Ubuntu to enable future researchers and developers to have a base upon which to further the work.

The VMWare image can also be used to generate a LiveCD to enable a cluster to be built with the minimum of effort.

1.4 Research Map



Hypothesis

Is it possible to facilitate the timely handling of large-scale digital evidence for professional computer forensic investigations, whilst still maintaining an appropriate chain of evidence, through the design of a suitable acquisition and processing methodology, implemented within a distributed middleware architecture?

This research aims

1. To derive a set of requirements to enable the development of a distributed management system specifically suited to forensic investigation. See 8.7.1;
2. To evaluate some prominent existing distributed management systems and assess their suitability to implement a prototype distributed forensic system. See 8.7.2;
3. To classify existing forensic investigation tools and assess the likelihood of running them in a distributed environment and if necessary, to derive a standard for new tools intended to run within a distributed environment. See 8.7.3;
4. To develop a robust design of a middleware framework to support processing digital forensic tools in a distributed environment. See 8.7.4;
5. To evaluate the prototype system using representative case data. See 8.7.5.

1.5 Thesis Structure

This section contains a roadmap to the rest of the document. During the research phase, it became apparent that this subject was multi-disciplinary. It draws together threads from a number of initially diverse subject areas and the chain of thought is not always clear. The next 7 sections describe the focus of the full section and offer a description of the contents.

1.5.1 Chapter 1 – Introduction

Chapter 1 introduced the subject area and covered that background to the problem. The case was made for the research strategy and the motivation, goals and objectives were described.

1.5.2 Chapter 2 – Research Methodology

In chapter 2 we present an extensive justification of our research methodology. We use a framework proposed by Ferris to select a methodology based on a scorecard system. Having chosen a Design method, we describe the process and consider the opportunities and risks associated with this technique. Agile is selected as a design method framework

1.5.3 Chapter 3 – A Background to Forensic Soundness

We start by considering Forensic Soundness in a general form, as it is implemented in both Analogue and Digital environments. Procedures considered as to be best practice are not a static set of rules. They have evolved over the last century. They continue to evolve, and have been influenced by thinking in domains such as quality assurance in the last 50 years. They are derived from the competing interests of all those connected with the investigation, the evidence and the calibre of the proceedings in legal procedure and so we will, briefly, place the current dilemma in the context of history and introduce the stakeholders. No solution can be considered complete without an understanding of the needs of the stakeholders. We become more specific in our definition of what actions can be taken to provide and increase Assurance and we assess the applicability of existing standards such as ISO 17025 and 27037, and how they are applied in this domain.

1.5.4 Chapter 4 – The Technical Perspective

In chapter 4, we explain the nature of the disruptive forces over the last 10 years and how they have caused the need for change. We describe the various solutions to these problems and conclude that Distributed Processing is the most promising.

We then assess some existing Distributed Systems for suitability in processing digital forensics. We suggest that in the next few years the current practice of ‘imaging the entire media’ will no longer be sustainable. We introduce a solution that moves away from ‘The Image’ to adopt Digital Evidence Containers but conclude that these will not attain the levels of assurance that have been established over the last two decades without full support from the operating system

1.5.5 Chapter 5 – Designing an Extensible Digital Forensic Investigations Solution

In our Design criteria, we explain the thought process upon which we base our design. We identify 18 specific problems and provide a brief explanation of the reasons for the problem. We conclude chapter 5 with a brief outline of the solution.

1.5.6 Chapter 6 – FCluster and Components

Chapter 6 introduces our solution FCluster and its component parts, Jigsaw Imaging, Prioritisation, FClusterfs. Jigsaw imaging is a novel non-linear imaging process that feeds the distributed processing cluster. We describe, in detail, FClusterfs, the middleware FUSE file system then how FClusterfs is the controlling force behind the access, movement and processing of data in an Assured environment

1.5.7 Chapter 7 - Implementation and the Test Rigs

In chapter 7, we describe the development environment and the test/evaluation environment that developed as a result. We also describe our selection of test data.

1.5.8 Chapter 8 - Evaluation

In Chapter 8, we undertake an evaluation of our proposal. We reject a numeric approach, choosing instead a qualitative evaluation of FCluster against a number of measures. Firstly, against our own criteria set out in chapter 4. We evaluate against the ACPO guidelines v5 and ISO 27037. We identified a paper by Daniel Ayers as a key document in our background in chapter 3. We consider our design against Daniel Ayers criteria. Finally, we evaluate as a candidate for compliance with OAIS ISO 14721.

1.5.9 Chapter 9 – Conclusions and Further Work

In chapter 9, we draw together all the previous work into a summary and identify gaps and possibilities for future work.

2 Research Methodology

2.1 Introduction

This research project is primarily concerned with the technical aspects of high performance processing of digital evidence but the solution must be recognised in the context of a larger system. The scope of the project has been clearly defined setting the boundaries of the project work but it would be negligent to completely ignore the wider implications.

The core of the work is to design, build and evaluate a system specifically tailored to process digital information in a manner that is acceptable to the ultimate users – the legal profession.

It is all too common for real world IT projects to fail. A report in 2011, produced by Oxford University in conjunction the McKinnsey Consultancy (Budzier & Flyvbjerg 2011), reports time overruns in 50% of large IT projects and budget overruns in 7%. Although the report accepts figures are highly questionable, being supplied by voluntary participation, it does contain figures of complete cancellation rates of 25%. The report highlights so called “Optimism Bias and Black Swan Blindness” where unexpected events have a major impact and it is argued, with hindsight, that they could have been foreseen if greater care had been taken. In a derived ‘Feature Article’ from the McKinnsey Consultancy ‘Software’ projects are often prone to 80% benefits shortfall over ‘non-Software’ projects. When IT executives were asked for their thoughts on the reasons for the failures ‘Missing Focus’, ‘Unclear Objectives, and ‘Lack of Business Focus’ were collectively rated highest at 30%. An ‘Unrealistic Schedule’ featured in about 25%. ‘Content Issues’ of ’Shifting Requirements’ and ‘Technical Complexity’ were next at about 20%. ‘Skills Issues’ were next with about 13%.

With this in mind, it would be prudent to at least attempt to plan this project in some detail. Clearly, from the above ‘Missing Focus’ is highly significant in the success of any project. It is now common for large organisations to have a ‘mission statement’ to act as a focus and guiding light to maintain the direction of the enterprise.

2.2 The selection of an appropriate methodology

In “On the Methods of Research for Systems Engineering” (Ferris 2009), Timothy Ferris draws on classical philosophy to derive a series of questions to help in the selection of a research method of a particular project.

In Table 1- Ferris’ Questions (Ferris 2009), we work though the questions and provide answers.

Dimension	Questions	Answer
Meta-dimensional questions	What is the subject matter of the proposed project?	Digital Forensic Data Processing
	Why will the proposed project be done?	To contribute to the subject domain and so attain a PhD
	Who will do the proposed project?	This is the sole work of the candidate
	For whom will the proposed project be done?	For the examiners and the digital forensic community
	When are the results of the proposed project required?	The Thesis must be submitted by April 2015
	Where will the proposed project be done?	At the University of South Wales
Desiderata	**D1** - Is the proposed project intended to make a significant contribution to the theory of the field?	No.
	D2 - Is the proposed project intended to make a significant contribution to the practice of the field?	Yes
Relation to knowledge	**K1** - Is the knowledge expected in the proposed project primarily desired for its intrinsic value?	No. It is applied research intended to be used to create a real world product at a later date.
	K2 - Is the knowledge expected in the proposed project primarily desired for its instrumental value as means to achieve something else?	Yes
Person who benefits	**P1** - Is the primary beneficiary of knowledge expected in the proposed project the researcher?	No

Dimension	Questions	Answer
	P2 - Is the primary beneficiary of knowledge expected in the proposed project people other than the researcher?	Yes, but I might become a developer
View of certainty of knowledge	**C1** - Does the proposed project presuppose that the knowledge to be developed concerns matters which objectively exist?	Yes, it improves on existing system designs
	C2 - Does the proposed project presuppose that the knowledge to be developed concerns matters which are constructs of the community?	No
View of tradition	**T1** - Does the proposed project presuppose that the existing framework of the field should be used as a foundation?	There are principles, namely, ACPO, that need to be respected but it is likely that this will challenge existing frameworks as it will most likely change one paradigm; that of the need for a single, master Forensic image.
	T2 - Does the proposed project presuppose that the existing framework of the field should be rejected or vigorously challenged?	No

Table 1- Ferris' Questions (Ferris 2009)

And so we get < D2, K2, P2, C1, T1 >, which from Ferris' work describes a Design Project.

> *"Design is an engineering research method in which the researcher addresses a problem which is important and novel through the activity of designing a solution. In a design research project the researcher finds means to solve the problem, thereby developing practice, and the knowledge developed is primarily developed for practical application with the possibility of some theoretical development as an additional outcome. The focus of design research is the development of knowledge which benefits others. The knowledge developed in design is certain because it concerns what does or does not work, in the context of a project seeking to deliver means to address a defined goal. The methods and knowledge used in design research are normally within the normal patterns established within the discipline, with the novelty being in either or both of the problem addressed or the exact combination of methods used to satisfy the objective."*

As part of the University of Alberta's CMPUT Teaching & Research Methods Course in 2007, Amaral et al. (Amaral 2007) write that there are several models that could be employed in Computing Science Research. Of methodologies listed in the paper, Formal, Experimental, Build, Process and Model, the "Build" approach is certainly nearest the spirit of Ferris' classification. Amaral describes it as:

> *"A "build" research methodology consists of building an artifact — either a physical artifact or a software system — to demonstrate that it is possible. To be considered research, the construction of the artifact must be new or it must include new features that have not been demonstrated before in other artifacts."*

It then goes on the further define the stages of a build methodology as

- Design the software System;
- Reuse components;
- Choose an adequate programming language;
- Consider testing all the time.

Amaral stresses the need for documentation and the need to "think before you build". He also suggests that the use of "text based data and communications formats" simplify testing and that "defining small interfaces increases flexibility and reuse potential".

In his course notes on research in Computing Science (Johnson n.d.), Chris Johnson calls this “Proof by demonstration” and suggests that it has much in common with accepted methods used in engineering practice. He points out “there are many reasons why this approach is an unsatisfactory model for research. The main objection is that it carries high risks”. This is most certainly true. This project is of a sufficient duration and ambition that it is not possible to foresee all eventualities and there could come a point at which an insurmountable problem, either because of complexity, resources or internal conflict, it is not possible to complete a prototype with sufficient functionality to be subjected to constructive evaluation. Johnson writes “The key problem here is that the iterative development of an artefact, in turn, requires a method or structure”.

2.3 The Design Science Research Framework

Adopting an established research framework would certainly help with attaining ‘Focus’ for this project. In “The Design Science Research Process: A Model For Producing And Presenting Information Systems Research” (Peffers et al. 2006), Ken Peffers et al. use the term Design Science Research to describe a newly emerging research methodology (new in 2006 when compared with far more established fields of research).

> *“We sought to design a design science research process (DSRP) model that would meet three objectives: it would be consistent with prior literature, it would provide a nominal process model for doing DS research, and it would provide a mental model for presenting and appreciating DS research in IS. The process includes six steps: problem identification and motivation, objectives for a solution, design and development, evaluation, and communication.”*

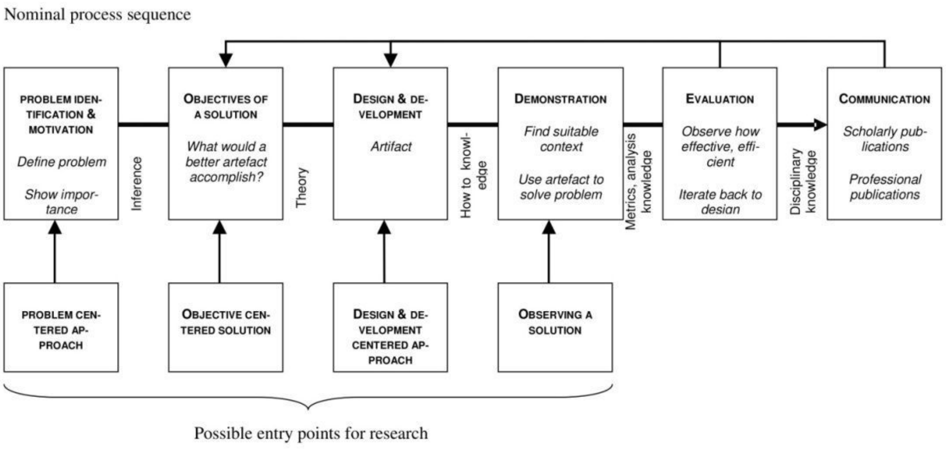


Figure 3 - DSRP framework

The DSRP model, shown in Figure 3, is highly abstracted and so needs elaboration for this project. If this project used the DSRP model, this project would initiate with the “Problem Centred Approach” entry point. This would be evident in those chapters within this thesis concerned with “Introduction” and “literature search”. Peffers does not specify at which stage a hypothesis is developed or at which stage an evaluation criterion is established. It does seem sensible if this was included in the “Objectives of a solution” model. Peffers has one large model “Design and Development”. This could be a single high risk area but it is included within a larger loop of “Iterate back to design”. Perhaps the model could benefit with greater emphasis on this iteration within the model. Peffers gives no indication of the rate for Design/Demonstration/Evaluation cycle, presumably leaving it to the individual researcher or project to dictate.

Alternatively, there is a model presented by Pillipp Offermann et al. at DESRIST in 2009 (Offermann et al. 2009) and shown in Figure 4. This develops on the same sources as Peffers. Offermann’s model but splits the literature search into two stages. The first is as part of the problem identification and the seconds is as part of the solution design stage. This is better in that it suggests that the design could be refined while in progress. This is certainly an advantage, if not a necessity, in a rapidly developing domain such as distributed processing.

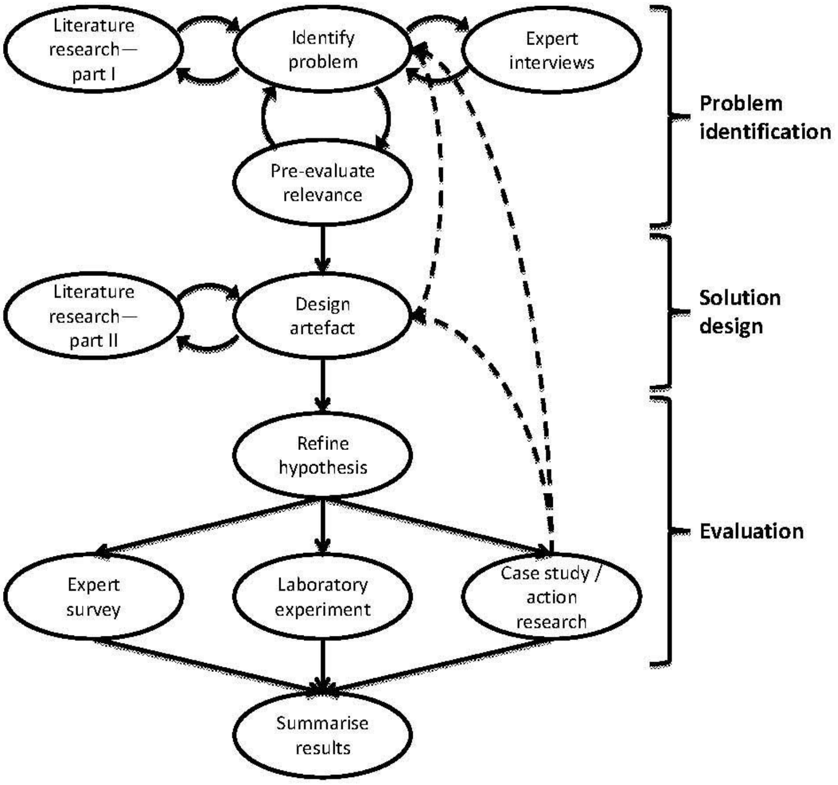


Figure 4 - Design Science Research Process

From “outline of a Design Science research Process”

This project will follow a fusion of these two models.

2.4 Risk management when building an artefact.

As Johnson identifies (Johnson n.d.) that when assessing what he calls “Proof By Demonstration” as a model for research - “However, there are many reasons why this approach is an unsatisfactory model for research. The main objection is that it carries high risks”. Having completed the initial literature research stage and an initial assessment of possible solutions, it seems likely that risk in this project could be managed, and so reduced, by breaking down the system into modules. For example, there is clearly a data acquisition stage and a data distribution stage. There are audit requirements and also system redundancy requirements. At this stage these seem to be independent of each other and could be designed, developed and evaluated independently using some type of software maturity model (Various 2013a). This implies some incremental approach to developing the individual modules. Perhaps each module could be the subject of independent development cycles, shown in Figure 5.

	Acquisition	Digital Evidence Bags	DEB storage	Batch Processing	Results Integration	Redundancy	Visualisation
Maturity Level				▨			
High			▨	▨			
Developed	▨		▨	▨			
Simplistic	▨	▨	▨	▨	▨	▨	
None	▨	▨	▨	▨	▨	▨	▨

Figure 5 - Project Module Maturity Model

Each of the units of work could be the subject of its own development cycle of Design, Build, Evaluate and Revue, shown in Figure 6. These loops are based on the Deming cycle of Plan, Do, Check Act (Various 2013b) introduced in the 1950s. This cyclical nature does not lend itself to heavy-weight programming project styles like Waterfall, introduced in the 1950s, which is characterised by a more rigid sequence of “Requirements”, “Design”, Implementation”, “Verification” and “Maintenance”.

2.4.1 Agile programming and project management

In recent years, a more dynamic, light-weight, style has been used in highly dynamic software project. Characterised by ‘Agile’, the approach is highly focused on producing a steady stream of valued outputs along the development line.

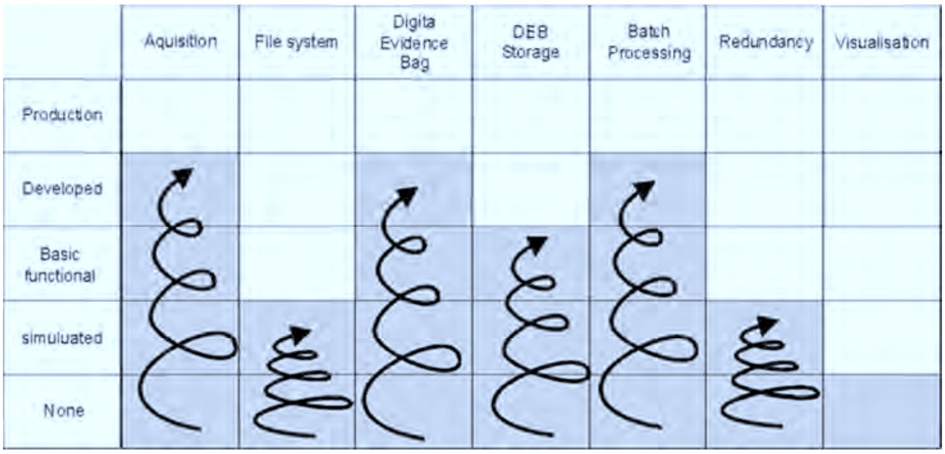


Figure 6 - Spiral Modules

2.4.2 The scale of the resources and the need for a simple, flexible framework

This may seem to be overly complex for the work of a single person but this is a project that represents 4,500 hours of work, a fixed financial budget, and a very specific timeframe. Agile is an ideal management model for these micro tasks.

2.4.3 Agile programming and The Agile Agenda as a research model

Agile has been evaluated as a model for research in Computer Science by Way, Chandraekhar and Murthy (Way et al. 2009). The family of Agile methods are based on iterative design processes that stand juxtaposed to the more rigid design and project management methods embodies in Waterfall and Prince2 for example. Although iterative approaches date back to the 1950s, Agile emerged in the late 1990s and is founded on the Agile Manifesto that was published in 2001 (Beedle 2001). There have been many popularised forms including Scrum and Extreme programming. Agile focuses on producing rapid development and adapting to a changing environment. Agile methodologies have a strong emphasis on continuous improvement. Way et al have wide experience of both industry and academic research and conclude that Agile has many feature well suited to academic research but there are some that are not. As a consequence they publish their "Agile Research Penultimatum". There are 12 principles adapted from the original template.

1. Our highest priority is to perform quality research through consistent effort and regular publication;

2. Welcome the unexpected, although better early than late in the process. Agile research processes enable early discovery, so care should be taken to minimize change later;

3. Maintain research documentation continually, updating a notebook or wiki as work is done and discoveries are made;

4. Faculty advisors and graduate students must meet weekly or regularly throughout the project;

5. Build projects around shared research goals that motivate faculty and student alike. Establish a work environment to support their individual needs, and trust them to get the job done;

6. The most efficient and effective method of conveying information to and within a research team is face-to-face conversation, but email, instant-messaging, wikis and blogs are essential media, as well;

7. Published papers, technical reports, literature surveys and working research software are the primary measures of progress;

8. Agile research processes adapt to the highly variable nature of the academic schedule, recognizing that the pace will vary dramatically over the course of a project;

9. Continuous attention to proper citation, short- and long-range planning, and maintaining forward momentum enhances research productivity;

10. Simplicity is a worthwhile goal, yet recognizing when complex problems often require complicated solutions is essential;

11. The best research emerges from self-motivated, highly-organized teams of one or more, yet chaos also can be an ally to discovery;

12. At regular intervals, individuals reflect on the effectiveness of their contribution to the team and adjust their behaviour accordingly;

Some do not apply to this project, as the team is just one person but most do.

2.5 Our finalised project strategy

The finalised project strategy and methodology is shown in Figure 7.

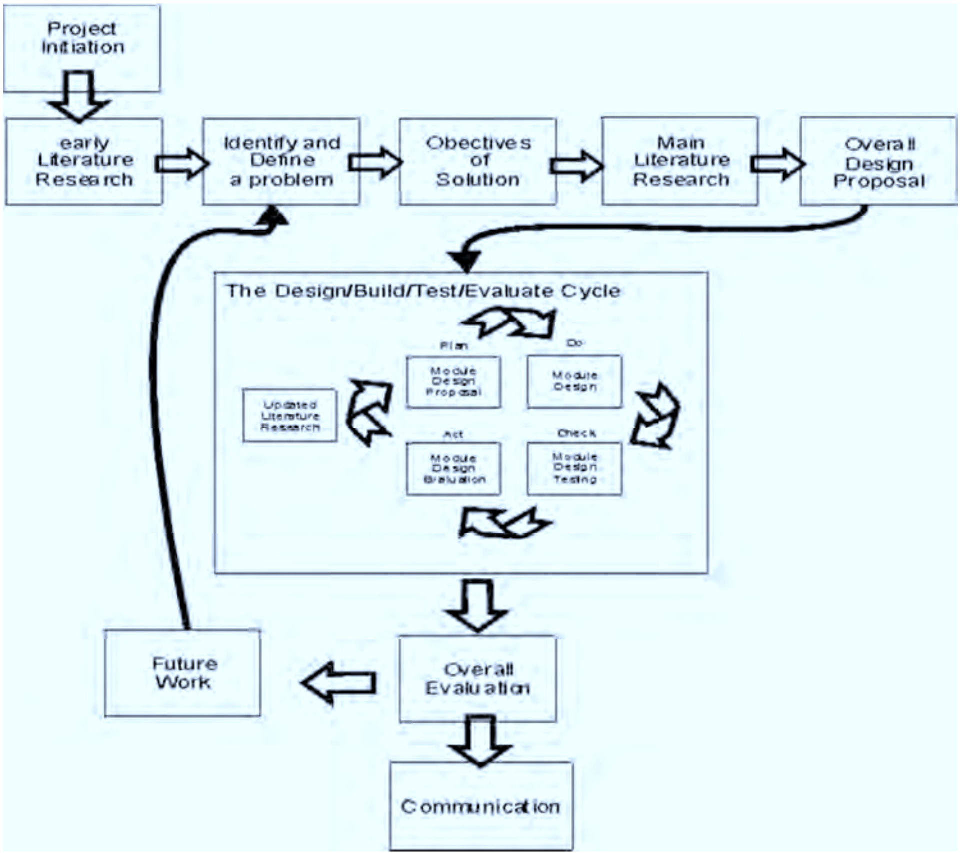


Figure 7 - Project Strategy

2.6 Conclusions

In chapter 2, we considered the issue of choosing an appropriate methodology for this research. Using a questionnaire, we were able to deduce that a design research methodology was most suitable. We then explored how this could be implemented and raised the issue of the risk of project failure and proposed a means of reducing that risk.

3 A Background to Forensic Soundness

In this chapter, we present the non-technical threads that come together to underpin this particular problem. We start by taking a broad view of forensic soundness in the analogue world. For more than 100 years, forensics has been applied to our analogue world. Suitable techniques have been developed to assure high standards of quality when processing analogue matter. There are however, significant differences when we consider the nature of digital information. Digital information can be copied and modified without detection. We consider the standards that have been developed over the last century, with particular emphasis on the last 10 years. These standards have been applied within digital forensics but are under strain because of the increase in the volume of data needing to be processed. We conclude this chapter with a consideration of the consequences on digital forensics' ability to deliver a quality server in the future.

3.1 Introduction

In the broadest sense, forensics can be said to be the study and presentation of information intended to, or at least assist in, proving or refuting an argument. A frequently used definition is that it "involves the preservation, identification, documentation and interpretation of computer data" (Kruse & Heiser 2001).

The Scientific Working Group on Digital Evidence (SWGDE) defined computer forensics as involving "the scientific examination, analysis, and/or evaluation of digital evidence in legal matters" (Scientific Working Group on Digital Evidence 2005). Our interpretation, within this text, is that it will be in the context of a legal framework.

3.2 Forensic Soundness in the Analogue World

Whereas digital forensics is a relatively new subject, perhaps only 20 years old, analogue forensics, has a history of more than a century. Therefore, we start the research into the characteristics of assurance within digital forensics by looking at the nature of analogue and digital evidence. We then turn to the standards and practices developed in the analogue world with a review of the legal view on the presentation of forensic evidence in court. These are based on standards and guidelines drawn up within the subject area and so we then review the relevant ISO standards. These standards are not static and so we outline the changes in thinking that have been used when writing the standards, namely the move from audit to assurance in the last 20 years.

3.2.1 The Analogue World

We live in an analogue universe. In the analogue world, there is no black and white, only shades of grey. Nothing is pure. Although concepts like absolute zero exist, it may never be achievable in the real world. In the real world, there is always error and randomness.

As an example of information in the analogue world, we could take a painting.

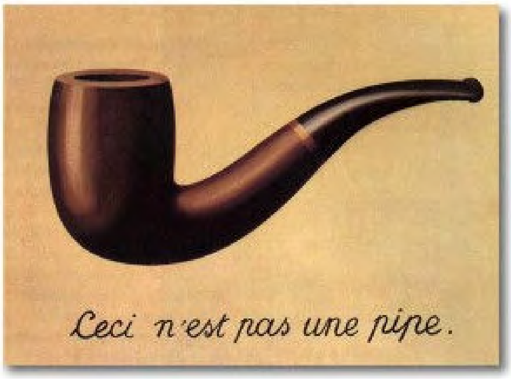


Figure 8 – The Treachery of Images - Rene Magritte – 1929

This famous painting by Rene Magritte, shown in Figure 8, (Magritte 1929) was used by Dan Farmer (2005) as the cover design for “Forensic Discovery”. He gives no explanation within his text as to his choice but clearly there must have been some thought process behind its selection.

Magritte was making the point that there is a higher meaning to the arrangement of oil on the canvas rather than just a pattern of colours and shades. It is not a pipe; it is a picture of a pipe but we take it as a pipe.

Magritte’s original is owned by the Los Angeles County Museum of Art and is unique. It is an oil painting on canvas. Prints have been made and are on display at many galleries around the world. No doubt, great care has been taken to reproduce the colours to the same as the original but none will succeed in a perfect reproduction in this analogue world. The composition of the oil used today will be slightly different to that used in 1929. It is possible to recreate something very similar but not identical. In the 80 years since the oil paint was made, mixed and applied to the canvas, which was also made in the 1920s, various chemical reactions have taken place. In effect, contamination has occurred. In fact, the colours in the original painting will have changed. It is likely that the original is stored in a climate-controlled environment that is intended to reduce to a minimum of chemical interaction within the oil. It would be expected that regular checks be made to ensure that the painting is not decaying beyond an expected boundary. In the analogue world, decay is inevitable; it cannot be stopped, only controlled. The analogue world is one with entropy.

3.2.2 Forensic Evidence in the Courts of England and United States

We now look at the subject from the view of the final customer, the legal system. From the late 1800s, increasing amounts of evidence derived from expert analysis was being presented in the English and American courts. As it became more common, a more formal framework was needed to assess the credibility and admissibility of the evidence and the methods behind its derivation.

The courts of the United Kingdom, The United states and number of other countries, usually ex-British Empire countries, are linked by their use of 'The Common Law' system. Within the Common Law judicial systems, that of the United States stands out because it has formalised the criteria for judging the admissibility of evidence into the proceedings. In a similar way as the United States has a written constitution and the United Kingdom has not, so the United States has a written code on the acceptance of evidence in court. The question of the quality of forensic evidence was raised in the United States nearly a century ago. Subsequently, we will assess the United States system first.

3.2.2.1 **Frye v. United States***, 293 F. 1013 (D.C. Cir. 1923).*

The ruling in Frye v. United States, in 1923, was concerned with the admissibility of polygraph tests as evidence shortly after their invention in 1921 (International League of Polygraph Examiners 2014). It established the president that evidence presented in court should be based upon scientific techniques. Although a number of States still adhere to the Frye standard, from 1923, most have been superseded by the adoption of the Daubert Standard in 1993.

3.2.2.2 The Daubert Standard

Daubert mostly clarifies the principle set out in Frye.

The Daubert standard provides a rule of evidence in the US courts based on three United States Supreme Court cases (Cornell University 2014).

- Daubert v. Merrell Dow Pharmaceuticals 509 U.S. 579 (1993), which held that the Rule 702 of the Federal Rules of Evidence, which accepted certain types of evidence, for example first hand testimony, as automatically admissible, did not apply to scientific expert testimony;
- General Electric Co v Joiner 522 U.S. 136 (1997) which held that district court judge may exclude evidence when there are gaps between the expert's evidence and the judge's own conclusions;
- That in Kumho Tire Co v Carmichael 526 U.S. 137 (1999) that the judge's discretion is applicable in all expert testimony including non-scientific evidence.

Daubert establishes the Judge as the gatekeeper on the admissibility of evidence and provides a non-exclusive checklist for trial courts to use in assessing the reliability of scientific expert testimony.

'Daubert' asks:

- Whether the expert is presenting evidence within their field of expertise;

- Whether the experts conclusions are a justifiable conclusion from their analysis;
- Whether the expert can account for alternative explanations;
- Whether the expert has been rigorous in their application of their expertise;
- Whether the field of expertise claimed is known to reach reliable results, “is it scientific”?
- Whether Empirical testing has been used, in that the technique is testable, refutable or falsifiable;
- Whether it has been subject to peer review;
- The degree to which the technique and theory is accepted by the relevant scientific community;
- Whether there is a known or potential error rate;
- Are there formal standards and controls concerning its operation.

Daubert does not attempt to set forth explicit procedural requirements for exercising the trial court's gatekeeping function over expert testimony. It makes no mention of how evidence is handled but it is reasonable to assume that the way in which evidential material is identified, collected and stored is part of ‘being rigorous’ in the application of the witness’s expertise.

It is obvious that a trial judge cannot be an expert in every field of evidence and so we are left in a situation where Forensic Soundness becomes an issue of ‘what is acceptable’ in a particular case as viewed by the judge. It is a question of what is ‘Best Practice’ in a field.

3.2.3 Evidence in the English Court

As in the American Court, the final decision as to the admissibility of expert evidence in the English court is made by the judge. However, there are a number of subtle differences in the English courts when compared with the American courts. Firstly, expert evidence is not admissible if it is on a subject that would be within the expected knowledge of the jurors. For example in R v Browning (1995 Crim LR 227) an expert was not allowed to testify on the matter of the expect deterioration of memory in ordinary healthy individuals as they age. However, expert evidence was accepted as to the reliability of memory from a subject under hypnosis (R v Land 1998 1 Cr App 301). The Daubert standard has been acknowledged in the English legal system but no similar standard has been formally applied. In R v Gilfoyle ([2001] 2 Cr App R 5) the court seemed to suggest that where expert evidence could not be independently reviewed by any given criteria it would be inadmissible. Later in R v Dallagher (2002 EWCA Crim 1903) this was expanded on and evidence was admitted despite not being widely accepted as reliable in the expert community.

The current situation seems to be that after R v Luttrell ([2004] 2 Cr App R 520) it is up to the judge to decide if the jury is capable of deciding the reliability of the expert and their testimony and, if so, allow the evidence with an appropriate caution about weighting.

It should be clear that when comparing the practices in the English and United States courts that no rigid rules have been created to dictate practices and procedures when analysing evidence for presentation in court. Instead, it is up to the judge to oversee reasonableness. No doubt, they look to established standards and relevant institutions for guidance in drawing up their criteria.

3.2.4 Standards for Analogue Forensics

In the UK, the governance of forensic services is overseen by the Forensic Science Regulator (FSR) (H M Government 2014). The FSR was created in 2008.

The FSR recently published the "Codes of Practice and Conduct" (Forensic Science Regulator 2011) for forensics expert evidence presented in the English court. The introduction of the Code of Conduct is driven by the European Union Council Framework Decision 2009/905/JHA on accreditation of forensic service providers carrying out laboratory activities concerning DNA analysis.

This is based upon a number of ISO standards and includes the notification that "Forensic Science Laboratories" in the UK will have to conform to ISO 17025:2005 "*General requirements for the competence of testing and calibration laboratories"* (ISO 2005) over next few years.

In his introduction, the head of the Authority, Andrew Rennison, does acknowledge that there is still debate as to the applicability of ISO 17025 to particular disciplines but clearly, he acknowledges it is a key document.

The focus on the use of ISO 17025:2005 seems largely driven by the need to attain conformance for the activities of scientific laboratories conducting work on analogue evidence such as firearms, fingerprints, DNA samples, marks and toxicology.

Because these standards need to be used in dynamic environments and, in the case of ISO standards, in a number of differing jurisdictions, none of these standards is prescriptive. They tend to use vague language like "competent" and rarely define exactly what that means. Section 5.2.1 – Human Factors, of ISO 17025 is a typical example.

> *"The laboratory management shall ensure the competence of all who operate specific equipment, perform tests and/or calibrations, evaluate results, and sign test reports and calibration certificates. When using staff who are undergoing training, appropriate supervision shall be provided. Personnel performing specific tasks shall be qualified on the basis of appropriate education, training, experience and/or demonstrated skills, as required."*

3.2.5 ISO 17025:2005

ISO 17025:2005 is often used as the basis for "Chain of Evidence" procedures in analogue forensics labs. This suggests that forensic laboratories are seen as places that conduct prescribed tests and produce a set of results. Although ISO 17025 is not prescriptive, it is useful to return to it to assess its scope. It includes:

- Human Factors (section 5.2 of ISO 17025) including competence, provision of on-going training, supervision, clearly defined tasks and responsibility;
- Accommodation and environmental conditions (section 5.3 of ISO 17025); the working environment should be adequate to support the analysis work undertaken. There should be environmental monitoring. There should be separation between areas that have incompatible activities. There should be good housekeeping in the laboratory;
- Test and calibration methods and method validation (section 5.4 of ISO 17025); There should be appropriate testing and calibration of equipment that meets the needs of the customers. These should be based on international standards and/or the manufacture's own standards and procedures. Where formal methods do not exist, new methods should be established and validated.

  Methods used in analysis should be selected to meet the needs of the customers and be of a suitably high quality. There is provision for non-standard tests, in that they must be validated first.
  - Selection of Methods (5.4.2 of ISO 17025)
  - Laboratory-developed methods (5.4.3 of ISO 17025)
  - Non-standard methods (5.4.4 of ISO 17025)
  - Validation of Methods (5.4.5 of ISO 17025)
  - Estimation of Uncertainty (5.4.6 of ISO 17025)
  - Control of Data (5.4.7 of ISO 17025);
- Equipment (section 5.5 of ISO 17025); the laboratory should have equipment which is at least adequate for the job. It should meet the required accuracy. Equipment should be uniquely identified. Equipment operated outside of the nominal levels should be subject to further conformance testing;
- Measurement traceability (section 5.6 of ISO 17025); Equipment should be calibrated as per the manufacturer's and International System of Units' instructions. The laboratory should use reference standards.
- Sampling (section 5.7 of ISO 17025); the laboratory should have a sampling plan;

- The handling of test and calibration items (section 5.8 of ISO 17025). Test and calibration sample should be handled and protected in a suitable manner;
- Assuring the quality of test and calibration results (5.9 of ISO 17025);
- Reporting the Results (5.10 of ISO 17025).

The words ‘assure’, ‘assuring’ or ‘assured’ appears a number of times in ISO 17025:2005. All ISO standards that describe management practices are formed around the concept of assurance as distinct from audit. It is important to clarify the difference.

3.2.6 From Audit to Assurance

During the 1960s, the Japanese introduced the idea of total quality assurance. The most important feature of this was that controls were introduced before an action took place, not after.

The dictionary definitions (Dictionary.com 2014) give a sense of the retrospective nature of an audit and the future intent of Assurance.

Audit (noun)

1. an official examination and verification of accounts and records, especially of financial accounts.

2. a report or statement reflecting an audit; a final statement of account.

Assurance (noun)

1. a positive declaration intended to give confidence; a promise.

synonyms: word of honour, word, guarantee, promise, pledge, vow, avowal, oath, bond, affirmation, undertaking, commitment

2 confidence or certainty in one's own abilities.

synonyms: self-confidence, confidence, self-assurance, belief in oneself, faith in oneself, positiveness, assertiveness, self-possession, self-reliance, nerve, poise, aplomb, presence of mind, phlegm, level-headedness, cool-headedness

Japanese production lines did not produce faulty goods because faulty components were not allowed to enter the production line. The effect of this change on the industrial base of the western world is a matter of history. During the 1970s and 1980s, products from Japan surged leaving their North American and European competition behind, being viewed as unreliable. Modern management systems like Total Quality Management and Six-Sigma have their focus on controlling inputs and processes during the manufacturing process. Increases in quality, and customer satisfaction, are natural consequences of this approach.

In its terms and definitions, 3.2.11, ISO 9001:2008 (ISO 2012c) defines quality assurance as "A part of quality management focused on providing confidence that quality requirements will be fulfilled".

Assurance would, more than likely use an audit as a means of establishing faith or belief in the system but an audit trail is not assurance.

3.2.7 Assurance as a 3-tier model

Assurance is often expressed as a three-tier model, Figure 9, taken from Military doctrine (Griffith 1963):

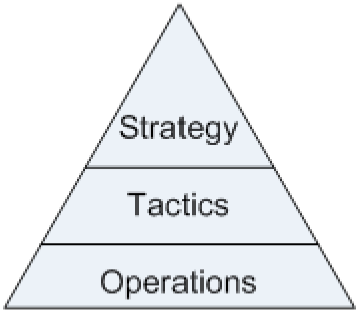


Figure 9 - Strategy Tactics and Operations

Strategy*: is the high-level plan: from the Greek stratēgía generalship, equivalent to stratēg (ós) military commander, general, a plan, method, or series of manoeuvres or stratagems for obtaining a specific goal or result:

Tactics* are a more specific implementation: from the Greek taktikós fit for arranging or ordering, implemented as one or more specific tasks, (used with a plural verb) the manoeuvres themselves.

Operations*: from the Latin, operātiōn- (stem of operātiō), equivalent to operāt (us) a process, method, or series of acts, especially of a practical or mechanical nature

Operations are more detailed than tactics, which in turn, are more detailed than strategy. ISO management standards focus on the strategic level and leave interpretation of this into tactics and operations to the implementer. This is simultaneously an advantage and disadvantage. It enables the advantage of flexibility for the implementer but at the same time lacks specifics upon which operations can be implemented.

Having failed to establish a clear criteria for assurance in analogue forensics we now complicate it further by revealing that there are significant differences between analogue and digital data.

3.3 Representing the analogue world in a digital form

As we saw, briefly in section 3.2.1, that the real world is analogue. The representation of the real world in a digital form is a human construct. There are many advantages in this representation. Digital data is black or white, ones or zeroes. Unlike analogue material, with appropriate safeguards and error checking, digital data can be replicated with the certainty that the replica is truly identical. Digital data is distinct from analogue material in that two items of digital data can exist in proximity without the risk of contamination.

We can illustrate the distinction between the storage of data in digital form and the information it represents by returning to the work by Magritte.

Figure 10 - Reworking of “The Treachery of Images”, is clearly not an exact facsimile of the original but does convey the same meaning. The colour has been removed and is probably a slightly different size. The data is now different but the meaning remains.

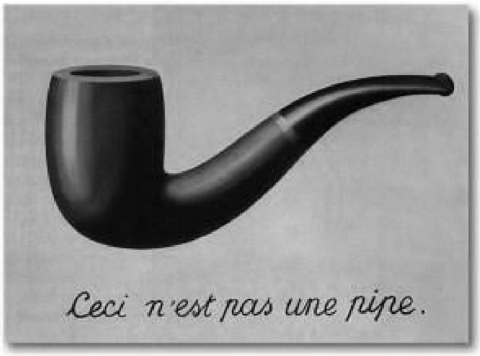


Figure 10 - Reworking of “The Treachery of Images”

Ironically, these images now display another aspect to the information content that Magritte could never have realised at the time of painting but may have seen towards the end of his life in 1967. As a necessity of the production of this document on a word processor, both of these images have been stored in a digital form. As such, they can be reproduced endlessly without any change at all to the digital pattern. Both files, the digital representations of the images above, co-exist on the surface of this computer, within this document and will be on any computer that is used to read this document. Each copy is true to the original digital version. It must be noted that although the original digital file will remain unchanged, the exact reproduction of both files, on the screen or paper, is dependent on the software used to decode the file.

It was easy to change the file and consequently the image. The original file was loaded into a graphics package, a few buttons were clicked and the results saved. If we chose, we could try to convince someone that the result was the picture painted by Magritte. There is normally nothing in the JPG file format to record that changes have been made to an original.

When digitising a continuous analogue source, there will always be a degree of error because of an effect called quantisation. However, once made, the digital version can be reproduced and stored endlessly without any reduction in its integrity. The question of what is original has deepened with David Hockney's recent work on his iPad (Hockney 2015).

The understanding that the nature of digital data is in some ways fundamentally different to analogue data implies the model of assurance drawn-up for analogue forensic needs to be different too.

The differences in the fundamental characteristics of digital when compared with analogue means that some criteria that have been applied to build assurance in analogue forensics do not apply to digital forensics, some extra criteria are needed in digital forensics and some remain present in both.

3.4 Confidentiality, Integrity and Availability

The notions of Confidentiality, Integrity and Availability (CIA) are often used as the founding principles of information security even ISO 27001:2013 (ISO 2013) defines Information Security as "preservation of confidentiality, integrity and availability of information". Its components are defined individually as part of ISO 27001:2013's definitions of terms.

There are seven relevant definitions with ISO 27001:2013.

3.4.1 Confidentiality

The definition of confidentiality is derived from ISO/IEC 7498-2 (ISO 2014) to be the "Property that information is not made available or disclosed to unauthorized individuals, entities, or processes". Confidentiality is typically provided by encryption and control over access to the data throughout its life cycle. To what degree access control is enforced, depends on the nature of the data, the environment in which the data is acquired and stored and the nature of the authority given to individuals. In section 1.3.5, our aims and objectives, the usage environment was set as a regional crime forensic lab with about 100-host cluster. This architecture may be extended off site to include cloud services and could include a facility to allow remote access to the system. Subsequently encryption standards should be strong, typically AES-256.

3.4.2 Integrity

Integrity is derived from ISO 13335-1:2004 (ISO 2004) where it is defined as the "property of safeguarding the accuracy and completeness of assets". This is, perhaps, the most important in our quest for an acceptable solution to support forensic processing in a distributed environment. It is essential that the original file remains unchanged from acquisition to processing. This is usually achieved by applying combinations of MD-5, SHA-1 and SHA-256 crypto hashes.

3.4.3 Availability

Availability is derived from ISO/IEC 7498-2 as “Being the property of being accessible and usable upon demand by an authorized entity”. Within the context of this thesis, this is mostly focused on system failure and redundancy. If the likelihood of the failure of a single device is said to be f_1 then the likelihood of a failure occurring in n devices is the product of the $f_1 \ldots f_{n.}$ As n increases so does the chance of a failure. In a large cluster, failure of individual components is almost certain and so it is common practice to address failure by replication of resources.

3.4.4 Non-Repudiation

The CIA acronym has been extended to include a new principle – non-repudiation. ISO 27001’s definition is

“[The] ability to prove the occurrence of a claimed event [..] or action and its originating entities, in order to resolve disputes about the occurrence or non-occurrence of the event [..] or action and involvement of entities in the event [..]”

In a system intended for forensic analysis this could mean that logging needs to be implemented.

Three more definitions within ISO 270001 are relevant:

3.4.5 An Asset

The previous definitions refer to the properties on an asset. ISO 27001’s definition of an asset is “anything that has value to the organization

NOTE there are many types of assets, including:

a) information (2.18);
b) software, such as a computer program;
c) physical, such as computer;
d) services;
e) people, and their qualifications, skills, and experience; and
f) intangibles, such as reputation and image.”

3.4.6 An Event

ISO 27001’s definition is 2.15 - “[An] occurrence of a particular set of circumstances”.

3.4.7 Information

ISO 27001’s definition is “knowledge or data that has value to the organization”. This is included to highlight the difference between raw data and the results of processing that data to produce useful information.

Almost all, more complete concepts in information security and assurance can be measured against these definitions.

3.5 Current Standards and Practices applicable in the UK

The development of standards and best practices is an organic process that, over time, represents the changing thoughts and environment in which the activity, in our case digital forensics, takes place. In this section, we review the output of various authorities concerned with the collection, analysis and presentation of digital evidence in a Court. This is a significantly large field in itself and so we focus on literature directly effecting UK.

3.5.1 Timeline of documents relevant to the UK

Table 2 contains a chronological listing of the key legislation, guidelines and standards published in the UK that are applicable to digital forensics. A number of events or publications are included in the table to place the key documents into an historic perspective.

Year	Organisation	Document	Notes
1970	Union Dime Savings Bank	First Computer Crime?	*included only for background
1984	Louisiana State Police	Computer Incident CERT	*included only for background
1990	UK Gov	Computer Misuse Act	*included only for background
1998	ACPO	Good Practice Guidelines v1	
1998	UK Gov	Data Protection Act	*included only for background
1998	ISO	ISO/IEC 17020:1998	General criteria for the operation of various types of bodies performing inspection
1999	ISO	ISO/IEC 17025:1999	Testing and calibration labs
1988	UK Gov	The Human Rights Act	
2003	ACPO	Good Practice Guidelines v3.0	
2005	ISO	ISO/IEC 17025:2005	Testing and calibration labs - update
2005	ISO	ISO/IEC 27001:2005	First publication of Information security management Standard
2006	UK Gov	Amendments to the Computer Misuse Act 1990 in amended within the Police and Justice Act 2006	*included only for background
2007	ACPO	Good Practice Guidelines v4	

Year	Organisation	Document	Notes
2011	ACPO	Good Practice Guidelines v5	
2011	Forensic Science Regulator	Code of Practice and Conduct	
2012	ISO	ISO/IEC 27037:2012 (31st October)	First publication of Guidelines for identification, collection, acquisition of digital evidence
2013	ISO	ISO/IEC 27001:2013	Updated publication of Information security management Standard
2014	Forensic Science Regulator	Addition of the Appendix to the Codes of Practice and Conduct: Digital Forensic Services	

Table 2 - Standards and Legislation Directly Relevant to the UK

3.5.2 ACPO Guidelines

The ACPO Guidelines on Digital Evidence (ACPO 2014), now in its 5th version, has been considered to be the principle reference document concerning the handling of digital evidence in the UK. From the first version, it introduced four principles that should guide all activities and procedures in an investigation.

Principle 1: No action taken by law enforcement agencies, persons employed within those agencies or their agents should change data that may subsequently be relied upon in court.

Principle 2: In circumstances where a person finds it necessary to access original data, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions.

Principle 3: An audit trail or other record of all processes applied to digital evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result.

Principle 4: The person in charge of the investigation has overall responsibility for ensuring that the law and these principles are adhered to.

From Principle 1, that no change should change data, we normally derive a requirement that access to the data should be read-only as soon as possible. Principle 2 is typically applied to the actions of a DEFR when collecting digital evidence and a DE when examining a “closed” device such as a proprietary embedded device as both situations may require devices to be operated as a user as no facility may exist to take a forensic image. Principle 3 clearly mandates that an audit trail must be created and maintained to enable a third party to achieve the same result. This implies that an audit trail would be expected to include levels of detail such as date and time, investigators identification and the version numbers and even serial numbers of particular tools used. Principle 4 is typically applied to a case management decision to assess the credibility of a particular process, technique or device used during an investigation.

Although all four principles are relevant and important throughout an investigation, within the scope of this research and the subsequent design, principles 1 and 3 are particularly significant as they affect the design and operation of the equipment used rather than the general conduct of the operatives conducting the investigation.

Further, on the matter of the collection and storage of evidence, the Guidelines specify that

> *"In order to comply with the principles of digital evidence, wherever practicable, proportionate and relevant an image should be made of the device. This will ensure that the original data is preserved, enabling an independent third party to re-examine it and achieve the same result, as required by principle 3."*

and

> *"This may be a physical / logical block image of the entire device, or a logical file image containing partial or selective data (which may be captured as a result of a triage process). Investigators should use their professional judgement to endeavour to capture all relevant evidence if this approach is adopted."*

The practice of taking a digital forensic image was one of the first adopted within digital forensics. Unlike any analogue forensics, digital forensics has a unique capability of capturing an entire crime scene in one object. If properly collected and stored, this object is immutable and verifiable as an identical replica of the original.

Prior to version 5 in 2012, the ACPO guidelines did not refer to any other external standards. Version 5 was published in 2011, the same year as the establishment of the Forensic Science Regulator. It does include reference to ISO standards 17025 (ISO 2005) and 17020 (ISO 2012d) but only in relation to the work of external contractors, not the Police themselves.

3.5.3 ISO Standards

There are four ISO standards most frequently applied to digital forensics.

- ISO/IEC 17020:2012
  Conformity assessment -- Requirements for the operation of various types of bodies performing inspection;
- BS EN ISO/IEC 17025:2005
  General requirements for the competence of testing and calibration laboratories;
- BS ISO/IEC 27001:2013/BS 7799-2:2013 Information technology
  Security techniques. Information security management systems. Requirements;
- BS ISO/IEC 27037:2012
  Information technology. Security techniques. Guidelines for identification, collection, acquisition, and preservation of digital evidence.

3.5.3.1 ISO 17020:2012, ISO 17025:2005 and ISO 27001:2013

ISO 17020:2012 is a wide-ranging standard that covers many aspects of the management of organisations conducting inspections. It is concerned with the organisational and administrative aspects of the organisation providing a service.

ISO 17025:2005, "*General requirements for the competence of testing and calibration laboratories",* previously assessed in section 3.2.5, is specifically referenced in section 7.3.4 of the ACPO guide v5. It was written with intention of setting standards of operation for laboratories engaged in work with subjects in the analogue world, mainly chemistry and physics.

Although these standards are applicable, in the ACPO guide they are used only in reference to the operation of external forensic providers providing services to the Police service.

The inclusion of a reference to ISO 17025 within the ACPO Guidelines is an indication that the authors did accept the association of subject areas but its context in a section about external contractors could betray the guidelines author's inability to apply it to the specific domain of digital data.

ISO 27001:2013 is the ISO standard for the management of information systems. This is often used as a framework for the operation of digital forensic laboratories but contains nothing of assistance on matters of how to handle and process evidential data.

3.5.3.2 ISO 27037:2012

As a response to the difficulty of applying ISO 17025 to digital evidence, ISO 27037 (ISO 2012b) was published in 2012 to meet the specific needs of digital rather than analogue forensics.

ISO 27037's full title is "Information technology – Security techniques – Guidelines for **identification**, **collection**, **acquisition**, and **preservation** of digital evidence". It was published in 2012, and so, to some degree, supersedes parts of the version 5 of the ACPO Guidelines in the specific areas specified in the title. As an international standard, it does not deal with preparation, jurisdiction or other matter specifically pertaining to the UK Police, as does the ACPO guidelines. There are a number of areas of specific interest.

3.5.3.3 ISO 27037 components

Many components of ISO 27037 echo ideas already covered in previous standards such as ISO 17025 but specifically focus on the features of digital evidence. Here, in Table 3, we review the relevant sections and draw out information specific to digital investigations.

Auditability section 5.3.2	ISO 27037 states that: "It should be possible for an independent assessor or other authorized interested parties to evaluate the activities performed by a Digital Evidence First Responder (DEFR) and Digital evidence Specialist (DES). This will be made possible by appropriately documenting all actions taken. The DEFR and DES should be able to justify the decision-making process in selecting a given course of action. Processes performed by a DEFR and DES should be available for independent assessment to determine if an appropriate scientific method, technique or procedure was followed."
Repeatability section 5.3.3 and **Reproducibility** section 5.3.4	Are the same as in ISO 17025:2005, in that repeatability is the same action on the same equipment should produce the same result and reproducibility means the same action on different equipment should produce the same result.
Justifiability section 5.3.5	This requires that the operator must be able to justify their actions and choice of technique
Identification section 5.4.2	ISO 27037 makes an additional distinction in identification over ISO 17025 in that digital evidence needs to be identified in both its physical and its logical form. For example, both the physical media and the data stored upon it need separate identification.
Collection section 5.4.3 and **Acquisition** section 5.4.4	Following on from Identification, ISO 27037 makes a distinction between collection, which is the physical removal of a device, e.g. hard disk, or the acquisition of the data on the device by a suitable acquisition method. The most common is 'imaging'.

Preservation section 5.4.5 and 7.1.4	Having collected or acquired the evidence it must be preserved. This embodies the principle of Integrity from section 3.4.2. This is often achieved by calculating a cryptographic hash key for the data, either as a whole or in parts. ISO 27037 does not refer to any archive of digital evidence for long-term storage.
Chain of Custody section 6.1	"The chain of custody record is a document or series of related documents that details the chain of custody and records who was responsible for handling potential digital evidence, either in the form of digital data or other formats (such as paper notes)" The Chain of Custody record should contain the following information as a minimum. • Unique evidence identifier; • Who accessed the evidence and the time and location it took place; • Who checked the evidence in and out from the evidence preservation facility and when it happened; • Why the evidence was checked out (which case and the purpose) and the relevant authority, if applicable; • Any unavoidable changes to the potential digital evidence, as well as the name of the individual responsible therefore and the justification for the introduction of the change.
Prioritising Collection and **Acquisition** section 6.8	This largely addresses the volatile nature of some data in digital devices, for example the contents of RAM which will be lost if the power is removed. The section does acknowledge that "time may be a limiting factor during an investigation. In these cases, preference should be given to potential digital evidence identified as relevant to the specific incident." It is notable that the word 'triage' does not appear in the standard and the standard gives no further indication of methods or techniques to implement this prioritisation. The standard does not seem to address the need for prioritisation due to the length of time required to process the data. There is no mention of anything like the 'The Golden Hour' principle, where evidence found very early on in an investigation is of greater utility, found in many Police documents (National Police Improvement Agency 2007).
Preservation of Potential Digital Evidence section 6.9	This largely deals with the preservation environment in terms of physical contaminants like moisture and electromagnetic forces. For example, it specifically addresses the need to maintain power on some devices. It does not address the issues of long-term archival storage. In the US retention rules vary state by state but it is not uncommon for them to be unlimited for very serious crimes (The National Center of Victims of Crime 2014)

Transportation 6.9.4	The standard includes text concerning the physical transportation of evidence but does not give guidance on anything concerning the logical transportation (communication) of data via a network,
Documentation section 6.6	The standard states that every activity should be documented. This is focused on the crime scene.

Table 3 - Relevant Sections of ISO 27037

Within the Chain of Custody, it treats the "evidence preservation facility" as a single entity but common sense suggests that if this were an extensive facility, as would be the case in a regional facility, records would need to indicate storage in a subdivision of the facility.

ISO 27037 does not address any matter in the handling of data during processing and analysis. It therefore doesn't acknowledge that a computer system can have parts.

3.5.3.4 ISO 27037 stakeholders

The introduction of the term Stakeholders is often attributed to Freeman (1983). Freeman proposes that the term stakeholder supersedes the previously used term, stockholders, and represents an acknowledgement that the actions of an organisation affect a wider range of parties rather than just the direct owners.

It is surprising that ISO 27037 identifies only four stakeholders. These are all involved, first hand, in the acquisition and investigative process.

- Digital Evidence First Responders (DEFRs)
  This is the person that gathers the evidence. It could be an employee of a company, an outside consultant or a Law Enforcer. Importantly ISO 27037 implies a separation of function by identifying the Digital Evidence Specialist as a separate entity.

> *"individual who is authorized, trained and qualified to act first at an incident scene in performing digital evidence collection and acquisition with the responsibility for handling that evidence";*

- Digital Evidence Specialists (DESs)
  The DES is *superior* to a DEFR in that they are

> *"[an] individual who can carry out the tasks of a DEFR and has specialized knowledge, skills and abilities to handle a wide range of technical issues".*

The standard also refers to two additional stakeholders but give no other information about their skills or tasks.

- Incident Response Specialists;
- Forensic Laboratory Managers.

We can speculate that the former is a DEFR with additional knowledge of incident response for dynamic scenarios, for example, where a digital intrusion, hacking, is in progress and additional knowledge would aid decisions about whether to terminate a link of let it continue to gather further evidence. The latter is most likely an administrator who need have little of not knowledge of digital matters as their expertise is in management.

Good system design should surely acknowledge the views of all stakeholders and be derived from the system's place in its wider operational context. As a result, we feel it important to acknowledge the inclusion of five additional stakeholders we observe in the legal process and subsequently the processing of digital evidence, and so extend the stakeholders presented in ISO 27037 to include six more, in Table 4.

Legal process (Mediators)	The legal process, and those employed by the process, has expectations of the evidence presented to them. Most of these are embodied in the standards we have reviewed. The evidence that is presented must achieve the standards set, for example by ISO 27037.
The person in charge of the investigation	In the APCO Guidelines, 3.5.2, principle 4 states that the person in charge of the investigation must ensure that the ACPO principles are satisfied. This implies that any technique, process or equipment needs to have been proven to achieve a satisfactory level of quality assurance to be eligible for use in the investigation.
Accuser	Often overlooked, the accuser has a right to privacy in matter other than those specifically associated with the accusation. Within digital forensics, this is most obvious in the surrender of media to the investigation that may contain personal data that has no relevance to the investigation or accusation.
Victims	The victims also have a right to expect their case to be heard in a timely manner and should expect respect in terms of their data required for the prosecution. In the UK this could be found to be embodied in the Human Rights Act 1998 (H M Goverment 2014).

The Accused – Suspects	In a similar sense, the accused has a reasonable right to privacy in the evidence they surrender. Probably because of a written constitution, US Law has the highest affirmation of a right to privacy (Electronic Frontier Foundation 2014).
Society in General (Customers)	Society in general has a right to expect the whole proceedings to be carried out in an efficient manner that is cost effective but still embodies the standards and principles of society.

Table 4 - Extended list of Stakeholders

Inclusion of these stakeholder interests somewhat extends the criteria for a forensic system. Notably, the ACPO v5 Guidelines do not refer to personal data privacy explicitly. It does however mention the European Convention of Human Rights Act,

> *"Due regard should also be taken concerning any possible contravention of the European Convention of Human Rights."*

which does have privacy as a basic right as article 8. Article 8 provides a right to respect for one's

> *"private and family life, his home and his correspondence", subject to certain restrictions that are "in accordance with law" and "necessary in a democratic society".*

3.5.4 The Human Rights Act 1988 and the Fourth Amendment to the United States Constitution

The Human Rights Act came into force in the UK on the 2nd October 2000. As previously highlighted, it is now acknowledged in the latest version of the ACPO Guidelines.

This parallels the meaning of Fourth Amendment to the United State Constitution that prohibits unreasonable searched and seizures. Over the brief history of digital forensics, it has been common practice for law enforcement officers to be able to acquire a complete image of the storage media associated with a case. The creep of the acceptance of digital storage in the life of ordinary people has been so rapid that the issue of privacy in this matter has only just started to be realised. Challenges to this practice are surely to be expected in the UK. In the United Sates Courts, Riley v. California (2014) the Supreme Court ruled that earlier Supreme Court decisions permitting searches incident to an arrest without a warrant do not apply to modern cell-phones as they are pervasive in our lives and contain a digital record of nearly every aspect of our lives. It may be that this restriction will be made more constricting in the next few years, if not for the accused but perhaps for the victim and other witnesses. Any new architecture for forensic analysis systems must accommodate this requirement of selectivity.

3.5.5 The Forensic Science Regulator's Code of Conduct on Digital Forensic Services

The Forensic Science Regulator's Codes of Conduct version 1.0, published in 2011, obviously makes no mention of ISO 27037:2012 as it was published a year later. The original version 1.0 does not really address any of the issues of digital forensics and mainly focuses on the problems of analogue forensics.

It has since been extended with "Appendix: Digital Forensic Services" (Forensic Science Regulator 2014) which does refer to IS 27037:2012.

The appendix includes a 'Scope' as:

> *"This appendix covers digital forensics work only as it applies to the identification, capture, preservation, investigation, evaluation, reporting and storage of data on digital data storage devices and mobile phone devices."*

Within the Code of Practice, section 6.1.2 says

> *"The [digital forensic service] provider shall take account of the need for backup and redundancy when working on cases, to ensure that a single technical failure (e.g. a power loss or disk corruption) will not result in the loss of data on working copies."*

whereas the ACPO Guidelines make no mention of backup or the need for redundancy in equipment.

The establishment of the Forensic Science Regulator and the publication of its guidelines will presumably not replace the ACPO guidelines, as the ACPO document focuses more on operational matters at the acquisition stage of the investigation.

3.6 Current Implementation

It may seem strange to think of a forensic analysis as a business process but this is a very suitable way of managing the process. It may be high-tech but it still has to operate within an organisational environment. Subsequently, many characteristics of the business process are shared with general business management but as different businesses have their own priorities and focus, so digital forensics has its own. Information assurance is certainly high in the list of priorities. Early investigative process models were largely linear. This helped clarity and transparency. The more recent models have acknowledged the non-linear nature of the investigation. However, in reality our observance of the near ritual of taking a forensic image has large consigned us to the original linear practices.

3.6.1 Implementation within a typical digital forensic facility

It is often thought that the Police and other digital forensic service providers have some special equipment not available to the general population. Although we acknowledge that some agencies will have equipment that is restricted to certain users, the equipment used by most facilities would be familiar to most ordinary computer users.

We believe the characteristics of our local Police service, Gwent Police, are typical of those of the 43 constabularies in the UK. Gwent Police have about 1,300 Officers who serve a population of about 560,000 over 1,500 sq kms. Gwent is an industrial area with one city, Newport, and several slightly smaller urban areas.

We know, from work with Gwent Police that they have a one Gigabit Ethernet network with about 15 PCs and a 70 Terabyte storage server.

From FOI requests, which are listed in the appendixes, we can see that they have seven specialists in the cybercrime unit and that they have been trained on EnCase from Guidance Software (Guidance Software 2014), FTK from AccessData (AccessData Corporation 2014), Cellebrite (Cellebrite 2014) and .Xry (Systemation 2015). They have a budget of about £65,000 pa, excluding salaries.

3.6.2 Business Controls in Digital Forensics

We have shown from the ISO Standards, the Code of Conduct from The Forensic Science Regulator and the ACPO guidelines that there are several sources of advice on how to form an assurance strategy for digital forensics. In assessing the true practical problems of implementing assurance in a distributed environment, we now turn our focus on the operations level to see how the controls can be designed and implemented.

> *"Management control can be defined as a systematic effort by business management to compare performance to predetermined standards, plans, or objectives in order to determine whether performance is in line with these standards and presumably in order to take any remedial action required to see that human and other corporate resources are being used in the most effective and efficient way possible in achieving corporate objectives."*

(Mockler 1970)

From conversations with officers in within our local Police service, we know that they keep track of work within their High-tech Crime Unit (HTCU) using a mixture of ink on paper and Excel spreadsheets. They record serial numbers on paper forms filled in with ink and require double signatures and further information is recorded on the Excel spreadsheets.

The reason for the difference is that the Excel Spreadsheets are seen as easy to administer but are vulnerable to amendment whereas ink on paper is, in practical terms, immutable. They have chosen a mix of the two to provide a balance between business efficiency on the one hand but non-repudiation on the other.

There are a number of attributes, or processes, that contribute to a sense of assurance within the realm of digital forensics. Many of these are shared with analogue forensics and, indeed, other activities like quality assurance. This has a surprising amount in common with the way a small business might operate.

In Table 5, on the next page, we list some typical “Business Controls” found in many business processes and some examples of their co-responding implementation in a forensic laboratory.

These are often based on a principle of “Checks and Balances”. The most commonly used in digital forensics is when a cryptographic hash is generated from a stream of data, the check, and the repeating of the this and comparison of the result to the value previously created and stored. If the two hashes ‘balance’, that is, they are identical, then the control is satisfied.

Business Controls in a forensic Laboratory - Table 5

Aspect	Description	Analogue/Business	Digital
By the Properties of an object:			
This identifies some inherent property that can periodically be re-assessed to confirm it has not changed. Ideally, this should be an immutable property or if not a property of which the change can be explained, for example evaporation would change the weight of an item.		Weight Physical Dimensions Colour Appearance A permanent serial number A suitably designed label is attached or is inside an evidence bag A unique mark or blemish that is not part of the original. For example, a cut on the sole of a shoe. *The physical device upon which digital data is stored exists within the analogue world.	Digital data held in the form of a file has a number of properties that can be used to ascertain its identity and integrity. • Its name! • A file obviously has a length. • A file has meta-data like creation date/time. • CRC – is a technique for detecting errors in digital data, but not for making corrections where errors are detected. CRC-12, CRC-16, CRC-CCITT, CRC-32. CRCs are subject to collisions because they are relatively short and so are not unique. 12 bit, 16 bit and 32 bit CRCs mean that for any data string there can only be 2^{32} = 4,294,967,296 results. They are typically be used to test the validity of chunks of data across transmissions lines. • Cryptographic hash like MD5, SHA-1 and SHA-2 family were once thought to be non-collision but MD5 and SHA-1 have been demonstrated to be broken at a technical level. SHA-2 [ref FIPS PUB 180-2] is now considered as a reliable standard.

Aspect	Description	Analogue/Business	Digital
By the Position/Location of the object:			
The fact that a file is in a certain location further enhances our faith that it is the correct one.		Held in a secure storage locker Is at a location which is unmovable e.g. a bend in a road	File data being part of a forensic image of some media obviously fits within the image as a whole.
By similarities and differences			
By cross testing for failure and success		Laboratory equipment is frequently tested for both success and failure to establish its conformance to design specifications. During these procedures, a series of sample are tested to see if the results are similar or different as are expected. In the analogue world, the results are rarely to be expected to be the same.	
By Loops of Authority and Acknowledgement:			
Proof that something was done by providing an order and a counter signed acknowledgement that it was actioned.		This could be in the form of instructions to a technician to perform a test on a sample. This could be written in ink on paper.	This could be a process triggered by the click of a mouse that when completed, generates a report back to the initiator containing information about the success or failure of the process and the results.

<table>
<tr><th>Aspect</th><th>Description</th><th>Analogue/Business</th><th>Digital</th></tr>
<tr><td colspan="4"><b>By Access control:</b></td></tr>
<tr><td colspan="2">By allowing and denying.</td><td>Where physical evidence is held in a secure storage locker with restricted access.</td><td>Using passwords, swipe cards. Write Blocker devices when creating a forensic image.</td></tr>
<tr><td colspan="4"><b>By Separation of process:</b></td></tr>
<tr><td colspan="2">having functionality provided by more than one program and clearly separating stages by function</td><td>First responder collected evidence. An analyst processes it. An investigator uses the information.
Separation of process within a lab so that contamination is prevented/controlled/minimised</td><td>The generally accepted practice of ring fencing the system. Control over installed software on the system
By separating DEFR and DES</td></tr>
<tr><td colspan="4"><b>By Audit trail:</b></td></tr>
<tr><td colspan="2">An historical record is kept for future reference.</td><td>This is achieved by recording actions taken in a form that is indelible or verifiable external to the task.</td><td>This is achieved by recording actions taken in a form that is indelible or verifiable external to the task.</td></tr>
<tr><td colspan="4"><b>By Checklist:</b></td></tr>
<tr><td colspan="2">By setting a list of actions or tasks that must be addressed</td><td>Checklists provide an aid-memoire to repetitive tasks to reduce error. These may be in the form of a tick list or as a set of workflow diagrams and procedures.</td><td>This is inherent in the procedural nature of a computer program.</td></tr>
</table>

Aspect	Description	Analogue/Business	Digital
By Training			
		As appropriate to the task.	As appropriate to the task.
By Calibration and testing			
		See ISO 17035	Attempts have been made to map ISO 27035 to digital forensics in the form of ISO 27037:2012.

Table 5 - Business Controls

3.6.3 The investigative process

The in late 1990s, being a relatively new subject, digital forensics needed to rapidly establish sound principles upon which to base future thinking about approaches, the creating of policies, forming procedures and establishing best practice. From this early thinking there a two legacies that remain as the basis of the subject.

Presumably, from the initial need for clarity, many of the yearly models of the investigative process were linear, or sequential, in nature. Pollitt (1995) proposed one of the first models shown in Figure 11.

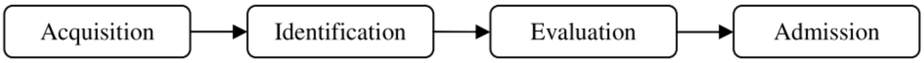


Figure 11 - Computer Forensic Investigative Process

The investigative process was initiated with the Acquisition stage. When acquisition was completed, it would be followed by the Identification, Evaluation and Admission phases, one after another. After one completed the next would start. At the time, this made sense. When an investigation often involved just one computer and subsequently one hard disk this was clear, clean, simple and so less prone to error. It was important that the process was clear and simple, as it may have to be justified to the court under the scrutiny of the Daubert Standard.

Six years later the 1st Digital Forensics Research Workshop (Palmer 2001) published its sequence, shown in Figure 12, with noticeable increases in the detail of early stages but still linear.

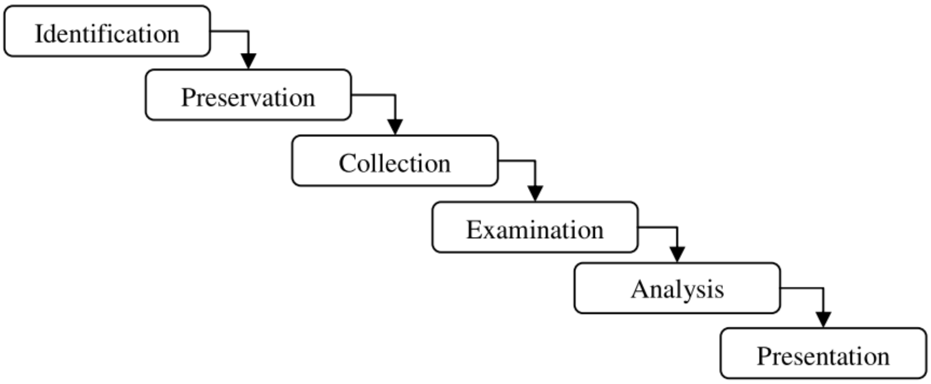


Figure 12 - DFRWS Investigative Model

The text within the paper actually describes it as linear.

> *“The items captured in [Table x] begin to establish the linear process, from Identification to Decision, that appears to be used in digital forensic analysis.”*

The text does elaborate on the potential activities within each stage, shown in Table 6.

Identification	**Preservation**	**Collection**	**Examination**	**Analysis**	**Presentation**	**Decision**
Event/Crime Detection	Case management	Preservation	Preservation	Preservation	Documentation	
Resolve Signature	Imaging techniques	Approved Methods	Traceability	Traceability	Expert Testimony	
Profile Detection	Chain of custody	Approved Software	Validation Techniques	Statistical	Clarification	
Anomalous Detection	Time Sync	Approved Hardware	Filtering Techniques	Protocols	Mission Impact Statement	
Complaints		Legal Authority	Pattern Matching	Data Mining	Recommended Counter Measure	
System Monitoring		Lossless Compression	Hidden Data Discovery	Timeline	Statistical Interpretation	
Audit Analysis		Sampling	Hidden Data Extraction	Link		
Etc.		Data Reduction		Spatial		
		Recovery Techniques				

Table 6 - Investigative Process for Digital Forensic Science (Gary Palmer 2001)

Later process models introduce loopback within the flow but it remains a largely linear sequence. The Abstract Digital Forensic Model (Reith 2002), shown in Figure 13, was the first to clearly display a loop-back which more realistically modelled the real world.

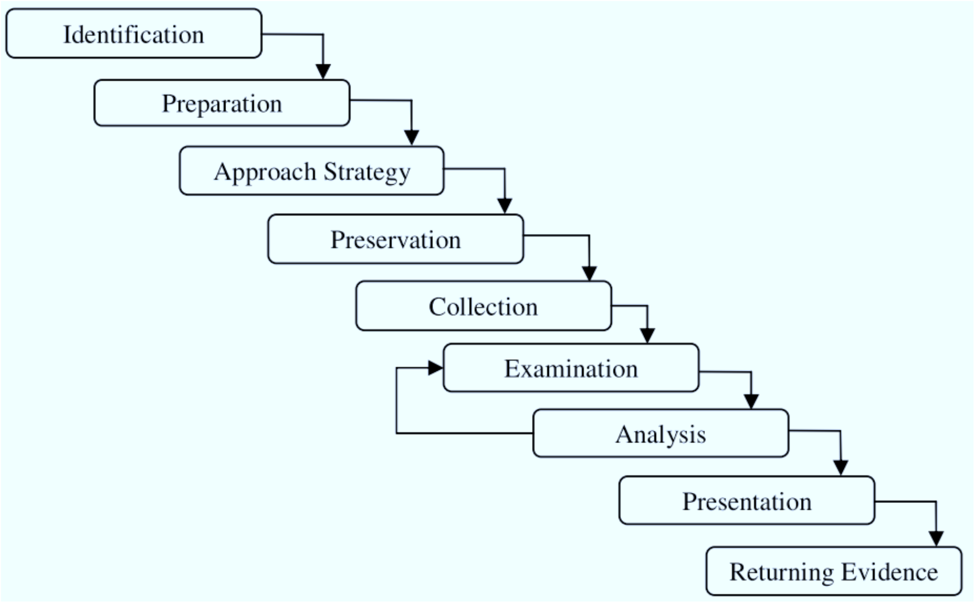


Figure 13 - Abstract Digital Forensic Model

The desire for simplicity and clarity is always a good principle but as, in many jurisdictions, the ultimate requirement of the forensic process is that the evidence is presented to a jury of laypersons, clarity and simplicity is essential to communicate the results of analysis.

3.6.4 Our reliance, or over-reliance, on the Forensic Image

Within the investigation, usually the preparation stage, the practice of imaging the media, inherited from the ‘dd’ program in Unix became one of the founding principles. Again, Forensic Imaging is a linear process and is easily understood.

From a chain of evidence or information assurance perspective, the practice of taking a forensic image in its current form is highly desirable.

The imaging process starts by acquiring storage block 0, then 1, then 2, and so on sequentially, to the last. This linear sequence also has the advantage that we can use a cryptographic hashing technique such as MD5 or SHA-1/256 to record the ‘state’ of the data at the time of recording and we are able to verify its integrity by repeating the process at any time and comparing the resulting hash, which should not have changed. This clear and simple process is robust and resilient to error or misuse.

Forensic imaging is also fast or last least, as fast as is possible. Typically, hard disks can deliver data at about 120MB/sec, this is limited by the rotational speed of the platter inside the device. A faster interface makes no different to the rotation speed and so does not increase the rate at which data can be delivered. The newer Solid State Devices, because of the absence of hardware that has been replaced by electronics, can deliver data at up to 500MB/sec.

Conventional forensic imaging is also a gentle process as it gathers data in a linear manner with the minimum of seek chatter that flicks the actuator arm mechanism side to side across the media. This reduces the possibility of hardware errors while imaging.

Having taken the image, the evidence, or crime scene, is considered to be preserved, and so it is safe. The image is something to which all things can be referenced. It is a datum point.

All this has combined to establish the practice of forensic imaging as a core activity in the investigative process. If members of the public know anything about digital forensics, it is likely to be about the practice of making a forensic image. Taking a forensic image is likely to be the first activity that students are told when starting to learn about forensics. It could be said to be in the DNA of the subject.

In the early days of digital forensics, imaging time could be expected to be one or, perhaps, two hours. It could usually be completed on site at the crime scene but was often left to be completed ‘back in the lab’ during the collection stage. This fell comfortably into the established investigative process models developed at the same time.

As media has become larger, this activity has taken longer and longer. In the next section, 3.7, 3V – Data Volume, Velocity and Variety, we will see the current trend in data volume and its consequence on imaging.

The technical aspects of forensic Imaging of media are more fully addressed in section 4.6.1.

3.7 3V – Data Volume, Velocity and Variety

During the first years, just after 2000, it seems as if researchers and investigators were making progress in developing techniques and tools that would enable effective investigation of digital information. However, in the same way as the rest of the world was rather surprised by the increase in the volume of digital data at the time, so the digital forensic community very soon found themselves in a situation where the ground attained in the previous decade seemed to be being lost to technical change.

In 2001 Doug Laney, an analyst working for the META Group, now the Gartner Group, described the growth of data he was observing as having three dimensions (Laney 2001). He was specifically referring to the accumulation of data in large organisation because of e-commerce but his observations are equally valid in the wider world. He observed data changing in three ways. Figure 14 demonstrates this graphically.

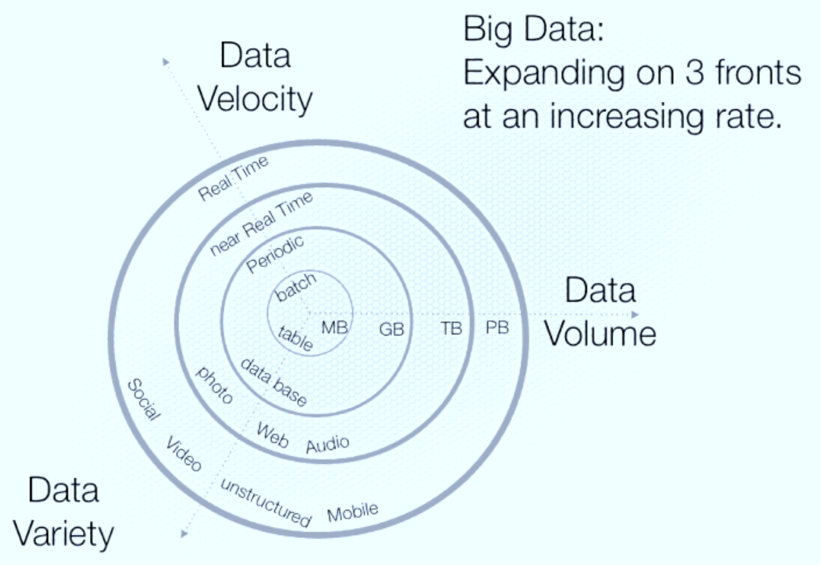


Figure 14 - 3V for Big Data (Soubra 2012)

3.7.1 Changes in Volume

Of the three changes, the most apparent is volume. Laney made his observations in 2001 when the use of the Internet by 'high street consumers' was in its infancy. The use of electronic point of sale devices was increasing. These devices, as well as dealing with monetary transactions were also able to capture data about sales and stock levels. In 2001 there was speculation that many organisations would soon like to adopt the 'capture and keep everything' approach to this data stream. Laney suggested solutions like this would result in huge increase in volume.

In another way, the increase in volume is clearly apparent when we remember that that in 1980, when the Apple II was on sale, a floppy disk was 160k and could store a 40-page document. Today, in 2014, a single digital photographic image, something that did not exist in 1980, will be more than 1000k.

3.7.2 Changes in Velocity

Laney also observed that there would be increases in the frequency of point-of-sale transactions. There would be an increase in electronic transactions via web sites and response speed on these sites would be a purchase differentiator. This has been realised as the desire the capture data in real time as it is generated at point of sale. Laney's work focused on commercial enterprise but this approach has been adopted across many domains. Within digital forensics, this is apparent in the number of cases being brought involving digital evidence. Fifteen years ago, digital evidence was rare. Today it is common.

3.7.3 Changes in Variety

The third expansion Doug Laney identified was variety. His observation was that many diverse and incompatible data structures would represent a problem for commerce, as data would need to be converted as it was passed between applications programs. In the forensic investigation domain, this translates into an ever-increasing variety of data sources that present as potential evidence targets. In 2001, Laney would have no idea that something called Social Media would become the force that it has.

3.8 3V in the real world

3V is all around us. There are three areas where it is very apparent.

3.8.1 UHDTV Video

One current example of data increase, from the consumer market, is the appearance of ultra-high definition video. In 2014, we are just seeing the appearance of 4k Ultra High Definition video. The specifications can be seen in Table 7 - HDTV data rates.

Name	Lines	Relative Pixels	Typical Data Rate	Typical Size of a Feature Film
HDTV	1080	1x	4 Mbytes/s	25 GB
4k UHDTV	2160	4x	16 Mbytes/s	100 GB
8k UHDTV	4320	16x	64 Mbytes/s	500 GB

Table 7 - HDTV data rates

What is notable, from Table 7, is that the data transfer rate is within the capabilities of current disk interfaces, local area networks and fast wide area broadband services. The notable increase is in the size of the data. A single layer Blue-Ray disk can hold 25GB, a dual layer 50GB and a BDXL 128GB. There is current no Blue-Ray disk specification that could hold an entire feature film in 8k UHTV. 4k broadcasts are happening now and 8k are due very soon. In November 2014, the BBC (2014c) reported that the Japanese broadcaster NHK is planning to offer 8K UHDTV for the summer Olympics in Tokyo in 2020. Soon, 4k will become a normal feature of domestic video equipment; it is already available in some equipment, like the GoPro Hero cameras.

The introduction of UHDTV is likely to drive storage capacities higher but may not have quite the same driving effect on data transfer speeds.

3.8.2 Cloud Computing and Storage

One of the major changes in computing in the last few years is the introduction of computing as a service and cloud computing. In essence, major computer resource users like Google, Microsoft and Amazon are renting out their facilities on a per unit basis. This has resulted in a number of well-known services like Google Docs, Microsoft SkyDrive and DropBox offering storage on their systems on a capacity/time cost basis.

It seems that, in effect, unlimited storage is now available and surely, usage will follow.

3.8.3 The Internet of things

Driven by the advent of System-On-a-Chip devices (SoC), more and more small devices, typically domestic white goods, will have local processing. When this is combined with Cloud storage and ubiquitous fast Internet access, it seems likely that a client-server approach will be increasing adopted for many personal computing applications. Many users, in all market sectors are choosing to have their data stored away from their host device with cloud storage services. This significantly changes the environment that DEFR will encounter. 10 years ago, in 2004, it was likely that a digital crime scene might contain one or two PCs and perhaps a notebook computer. It was likely there would have been an Internet connection but it would most likely have been no faster than one Mbytes/s. Now, in 2014, it is more likely that they will encounter many small devices that, for the most part, have access to data, stored remotely, on ‘a need to access’ basis. In the near future there may be many, perhaps dozens of devices to consider as candidates for preservation and collection, all with a different data storage format.

3.9 The consequences for Digital Forensics

Digital forensics exists within a global environment. Data volume growth has had a noticeable impact upon the ability of investigators to complete their work. Case numbers and volumes have increased along with the multi-nationalisation of crime. This then leads on to the needs to share investigative resources across jurisdictions. The data volume prompts the need for more automation. The linear nature of our investigative process is under pressure and one of the Holy icons of the domain, the forensic image has found itself under threat as it is too time consuming and it blocks further productive work for being undertaken. Subsequently, we come to the main manifestation of the problem, in that universally we are experiencing a growing backlog of forensic work needing to be done.

3.9.1 Data Growth in Digital Forensic Analysis

Within digital forensics, data growth and the need for change was identified more than a decade ago; in “Breaking the Performance Wall” (Roussev & Richard III 2004). At the time, they gave an example of a 6 GB hard disk examination taking two hours and extrapolating that to what was at the time an unfeasibly large 200 GB and the 60 hours it would take to do the examination. Two years later, in 2006, they return to the issue in a more broad way in “Digital Forensics Tools: The Next Generation” (Richard & Roussev 2006), stating that “massive increases in storage capacity are on the horizon” including the word “terabytes” for the first time in the domain specific literature.

There are now real examples of the need to process large datasets. In the summer of 2006 the London Metropolitan Police revealed that as a consequence of the summer’s transatlantic terrorist bombing threat they had impounded 400 PCs, 200 mobile phones and 8,500 items of digital media presumably USB sticks, CDs, DVDs etc.,(BBC 2006). The statement went further to say that this totalled about 6TB of data; if so, they must have been old PCs and mostly CDs rather than DVDs. With new computers and DVDs, this could total 15-20TB. Late in 2006 the Department of Defence seized 60TB of data after a leak of Iraq battle plans (ComputerWorld 2005). During the summer of 2014 the BBC(BBC 2014a) reported on the UK’s National Crime Agency’s operation Notarise in which 660 arrests were made in connection with child abuse investigations. The NCA report “833 buildings searched and 9,172 devices, including phones and laptops”. In operation Big Wing (Metropolitan Police 2014), the Metropolitan Police arrested 630 suspects in the London area in connection with house burglaries with hundreds of mobile phones and laptop computers to analyse. This was part of a national operation and in Leeds in 2009 the Yorkshire Police arrested 75 people in connection with burglaries in the county (BBC 2009). There were, we assume, many more regional operations that were not reported in the media. In October of 2014 the BBC (BBC 2014b) reported on the results of a FOI request by the NSPCC to all 44 Police Constabularies in England and Wales about the number of hard disks assessed in the previous 12 month period. Twenty-seven Forces responded with reports from, for example, Lancashire, Hertfordshire and Avon and Somerset constabularies seizing respectively 745, 516 and 466 hard disks in the previous 12 months. They employed 3, 4 and 13 staff respectively. The BBC report suggests that the Avon and Somerset staff may not all be digital forensic staff. The London Metropolitan Police Forensic Unit did not reply to the request but an informed source suggests that they may have to assess as many as 10,000 disks in a year. As there are fifty Police forces in the UK (Home Office 2014), it is reasonable to assume that this might amount to about 40,000 disks in a year in the UK. The NSPCC FOI request was specifically about hard disks and probably does not include any other digital media.

These types of operations are becoming more common as Law Enforcement realise they can gain more intelligence information from the devices they seize and so seize mode devices.

In the United States, the Regional Crime Forensics Laboratories annual reports (FBI 2014b) confirm the scale of the increase with their data consolidated in Table 8.

Year	Service Requests Received	Examinations conducted	TB processed	Average case size (GB)
2003	1,444	987	82	83
2004	1,548	1,304	229	176
2005	3,434	2,977	457	154
2006	4,214	3,633	916	252
2007	4,567	4,634	1,288	278
2008	5,057	4,524	1,756	388
2009	5,616	6,016	2,334	388
2010	5,985	6,564	3,086	470
2011	6,318	7,629	4,263	559
2012	5,060	8,566	5,986	699
2013	6,040	7,273	5,973	821

Table 8 - RCFL Annual Reports 2003 – 2013 (FBI 2014b)

The trend becomes clearer when represented as in graphical form with a logarithmic scale, shown in Figure 15.

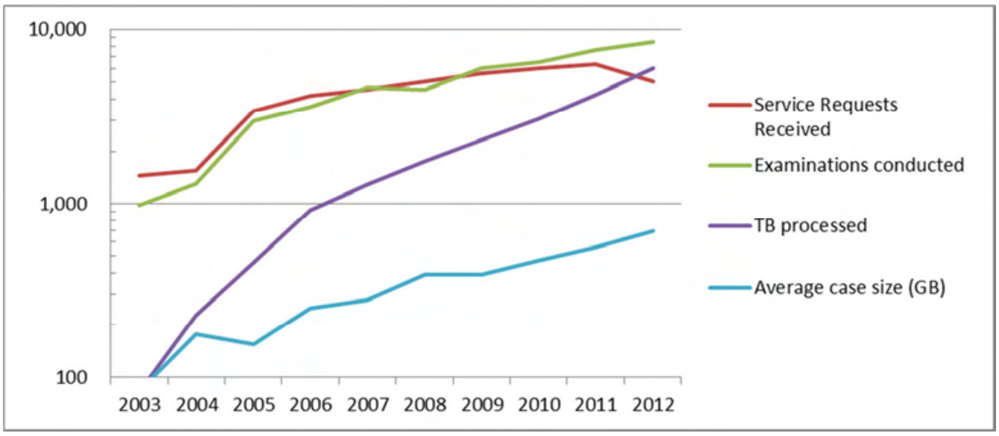


Figure 15 - RCFL Annual Reports 2003-2012 (FBI 2014b)

Garfinkel (Garfinkel 2010) makes a number of observations relevant to this research.

Firstly,

> *"Today much of the last decade's progress is quickly becoming irrelevant. Digital Forensics is facing a crisis. Hard-won capabilities are in jeopardy of being diminished or even lost as the result of advances and fundamental changes in the computer industry:"*

and observers that

> *"The growing size of storage devices means that there is frequently insufficient time to create a forensic image of a subject device, or to process all of the data once it is found."*

Garfinkel also introduces the idea of cross drive forensics or cases with multiple evidence items and the growing scale of the problem in this dimension.

> *“"Whereas cases were previously limited to the analysis of a single device, increasingly cases require the analysis of multiple devices followed by the correlation of the found evidence."*

and finally calls for realistic dataset when developing tools

> *"Scale is an important issue to address early in the research process. Today many techniques that are developed and demonstrated on relatively small data sets (n < 100) fail when they are scaled up to real-world sizes (n > 10,000). This is true whether n refers to the number of JPEGs, TB, hard drives or cell phones."*

There is more confirmation of the real world presence of the data size problem from Al Fahdi et al. (2013). In their survey of 42 researchers and practitioners, “Volume of Data” and “Time Taken” are identified as principle limitations to investigations. The respondents also identified the importance of automation to reduce workload.

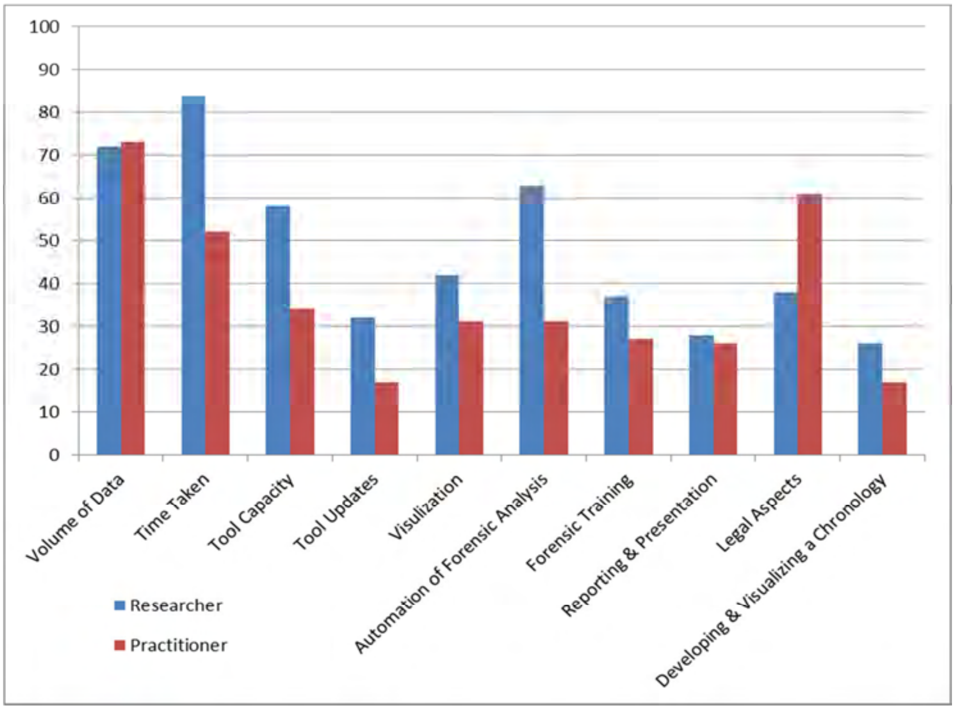


Figure 16 - Principle Limitations to Investigations (Al Fahdi et al. 2013)

The three categories on the left of the graph in Figure 16 are “Volume of Data”, “Time Taken” and “Tool Capability” the next highest is “Automation of Forensic Process”. These are all consequences of data growth.

3.9.2 Sharing evidence between agencies

The previous data gives an idea of the scale of the task but we also need to ask who will access the data? It seems likely that information may be of interest to more than one agency. The United Nations Cyber Crime Study 2013 (United Nations Office on Drugs & Crime 2013) devotes an entire chapter to “International Co-operation”. From the Study cybercrime questionnaire, Q83 more than half of the respondents reported that between 50 and 100 percent of cybercrime acts involve a ‘transnational element’ - Figure 17 - Transnational Component . We notice that the region with the largest reported value is Europe. We suspect that the other regions will soon catch up.

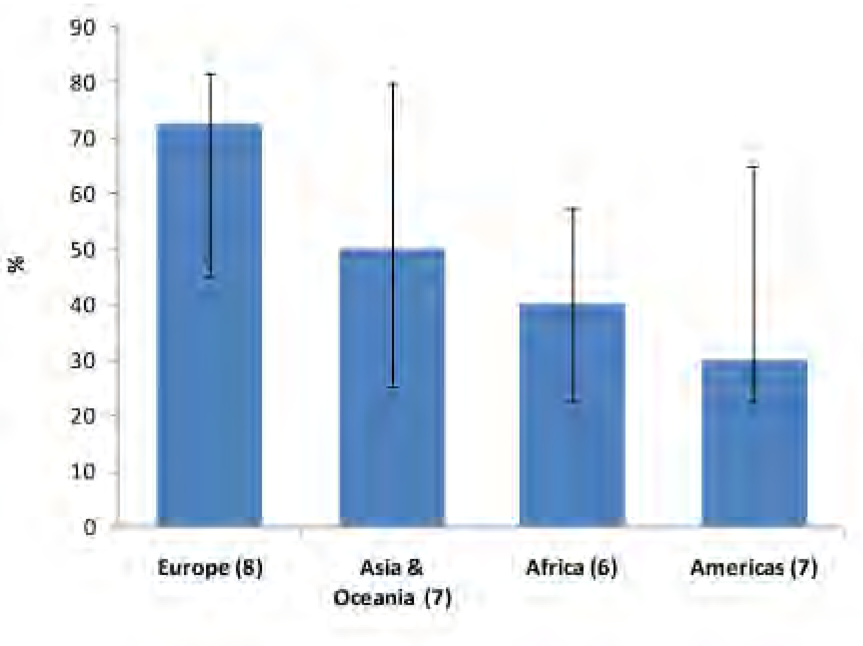


Figure 17 - Transnational Component. Source - UN Cybercrime Report

The consequence of this is that is that there will be an increase in the need to share either evidence or the results of analysis between investigative partners. These partners may not be within the same jurisdiction.

3.9.3 The need for Cross media Forensics

Perhaps the most noticeable trend in computing in the last 20 years is the sheer variety and number of devices that are now available. In the 1990s, in the UK, we might expect, perhaps, one desktop computer per household. Now, in 2014, there could be three or four digital devices per person in the same household. The ONS provide broad statistics for the last 10 years (Office of National Statistics 2012).

This trend will surely continue. “The Internet of Things” describes a world where previously dumb devices will contain processing power, storage and a connection to the Internet (Lyon 2013). It seems likely that in the future there may be many more devices that contribute to the evidence pool for a specific crime.

3.9.4 The need for Automated Analysis

We believe that automation will be a key part of the solution to processing the vast quantity of data needing analysis. The question is, to what degree automation will be used?

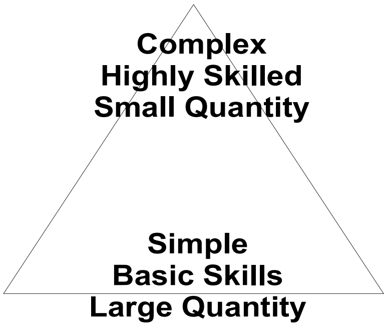


Figure 18 - Skill/Complexity/Quantity Pyramid

From our informal exchanges with Officers within the Gwent Digital forensics unit, we know they broadly believe that most of the information, or evidence, can be generated by a large number of simple actions each requiring relatively little effort. For example documenting thousands of pornographic images found on one hard disk. It is largely simple but very time consuming. They observe however, that the remaining evidence can take many times more effort to be retrieved by highly skilled officers. This is, perhaps, an example of the Pareto principle of 80/20 in that 20% of the effort generates 80% of the results. It then takes a disproportionately larger amount of effort to gain the remaining results. The principle is recursive in that 80% of the remaining effort is needed to gain the 20% of the remaining results and so on. A large amount of the work is simple, while a limited amount is highly complex and requires highly trained, skilled investigators, Figure 18.

3.9.5 The loss of ‘the image’

It is becoming more common to need to process multiple multi-terabyte devices. In 2015, with a 6TB drive, assuming no errors occur, imaging could take about 30 hours. To verify the image taken would require the process to be repeated taking the same time again. Only after the collection process is complete can any analysis begin. Roussev (2013) described this as resulting in forensics being viewed as “an open-ending, post-mortem analysis” that is fundamentally a sequential model”. These devices are becoming too large to be analysed as a whole.

We have been accustomed to using computers where the data is held on local storage devices, either usually hard drives or memory sticks formatted with familiar file-systems like FAT, NTFS and EXT3/4, or perhaps remote network drives which use the same technology. We now have to accept that the data may be stored in completely new formats more suited to the needs of the data-centre dealing with issues of load-balancing and power consumption, rather than those of the local operating system, and that these are made available via a network mount using protocols like CIF, SMB or HTTP. Perhaps they will be cached on the local host, perhaps not.

It seems that effectiveness of 'the image', which has been one of the foundations of forensic soundness, is being degraded and has a limited life expectancy. There will surely be a move to acquisition based on selective rather than whole device techniques.

The design and use of Digital Evidence Bags or Containers is a progression from the initial forensic image taken when preserving evidence. They are covered from a technical perspective in section 4.6.2, but here we place their development in a chronological context to highlight the evolution of features.

Phil Turner (2006) introduced the idea of a forensically sound unit of storage, smaller than a whole media image. "Selective Imaging" creates files called Digital Evidence Bags (DECs) which have a hierarchical structure. The design included extensible meta-tags. One of these groups is classified as *integrity* and could contain an MD5 or SHA cryptographic hash for each object within the DEC.

Turner's use of the term Bag was appropriate in 2006 as it made a clear distinction from the images of whole media in previous designed but now causes confusion. Digital Evidence Container (DEC) is a preferred term. Phil Turner does not appear to have continued this line of research.

Garfinkel (Garfinkel et al. 2006) introduced another DEC storage format – Advanced Forensic Format (AFF). It is delivered as an imaging command line program, a set of manipulation utilities and a set of C APIs. This was further extended by implementing a FUSE file system interface to the format with affuse. This allows non-AFF aware programs to access the raw data with an AFF file without 'knowing' anything about the AFF format. The FUSE file system code provides all the decoding functionality leaving the application program needing no amendments. FUSE file systems are covered in detail in section 4.5.

AFF has been adopted as one of the standard digital evidence container formats within the domain.

The bulk of Schatz's (Schatz 2007) work is focused on the structure of the information gained as a result of processing data using Formal Knowledge Representation but there is also a new proposal for a 'Sealable Digital Evidence Bag' based on Turner's design. Schatz uses a SHA1 digest in combination with meta-data about the source, acquisition tool and the investigator to record the integrity of the captured evidence. Schatz leaves the assurance of his evidence bags to further research.

Garfinkel (2009) returned to AFF, with version 3, and introduces four integrity features. There are three hashing and parity functions but, most importantly, there is a chain of evidence function. Files can be copied using an AFF function, afcopy that can have an X.509 pem file as a parameter.

Using Public Key Cryptography, the original is decrypted using the current private key and copies are 'issued' to the public key of the recipient. These can only be decrypted by the holder of the private key of the recipient pair. Evidence is unpackaged with the original cryptographic hashes and repackaged using the new. A record of the source history is included in the newly created DEC. Copy after copy adds more chain of evidence meta-data and so a chain of evidence can be established.

Cohen et al. (2009) developed his own major extension to the AFF format. Cohen refers to this as version 4, to mark a distinction from Garfinkel's branch of the development of AFF. Cohen's AFF4 is based on the original AFF format. In his section 8.1 "Using distributed Evidence" and 8.2 "Load Distribution" he alludes to the implementation of AFF in clusters and distributed processing but are not complete enough to fully understand how it would actually be implemented.

Richard et al. (2007) and Marziale (Marziale 2009) introduce the Forensic Discovery Auditing Module (FADM), which uses a FUSE file system to record all access to a file, thus providing an audit trail. This system creates a simple text file that records any file access, be it open, read, write or close.

Moving on to the procedural perspective of digital forensics, Casey (2013) addresses the issue of the accepted practices in the investigative process, which we have seen, are largely linear, and strives to "dismantle[s] the barriers between steps in prior digital investigation process models". He notes that

> *"{a] serial approach is not scalable because the process becomes less efficient as the size of storage media grows, particularly given the fact that disk I/O is the slowest operation in forensic processing."*

He suggests that

> *"A potential solution to this issue is to combine evidence acquisition with the automated extraction of information. Specifically, a forensic acquisition method can be augmented to simultaneously feed data into an extraction process while creating the forensic duplicate. Performing multiple extraction processes in parallel with forensic acquisition reduces the need to wait for the acquisition process to finish before performing further processing, thus increasing overall efficiency. In addition, to support further forensic examination of evidence without expending additional time making working copies, forensic duplicates can be stored on a*

> *network accessible storage system to provide all forensic processes with read-only access to acquired data in a single, centralized location."*

3.9.6 The loss of immediacy

In the UK, the National Police Improvement Agency (NPIA) published "Level 2 Guide to Evidencing Competency in Conducting Serious and Complex Investigations" titled "THE GOLDEN HOUR A competent investigator knows that time is crucial to an investigation" (National Police Improvement Agency 2007) as an expression of the need for urgent analysis, at least in serious and complex crimes. With increasing media size extending the time to take an image, we are moving further away from satisfying the need to collect and process data in a timely manner.

A solution could be to do the analysis first, directly from the source media, but this exposes the acquisition process to the accusation of tampering with the evidence before it is imaged. The ACPO guidelines state, "Wherever practical, an image should be made of the target device". Over the course of this research, Police officers have expressed a general concern with failing to conform to this requirement. Some, so much, that they have said that they would not present data collected in any other way than imaging as prima-facia evidence in court for fear that the cross examination of their methods could undermine the credibility of the rest of the prosecution case.

3.9.7 The Backlog

Clearly, the effect of this growth in data has serious consequences to the timely analysis of evidence for legal proceedings.

Vassil Roussev predicted this 10 years ago when he said "These trends will very soon completely overwhelm digital forensics investigators attempting investigations using a single workstation as a platform." (Roussev & Richard III 2004).

Harry Parsonage (Parsonage 2009) reported that "At present in 2009, it is commonplace for digital forensic units to have a backlog, several as long as twelve months."

Gogolin (Gogolin 2010) reports that the scale of the problem is growing exponentially with a statement

> *"Many law enforcement agencies reported during the interviews conducted that 50% or more of their cases have a digital component, and most agencies report that this number is increasing. Couple this with the fact that many digital crime labs, including the state digital crime labs in Michigan, have backlogs approaching or exceeding 2 years."*

Nina van der Knaap, of Leiden University, recently presented a work in progress report of her survey into work backlogs in forensic analysis at DFRWS Europe in Amsterdam in 2014 (van der Knaap 2013). She has run a small survey for 2 years and although the general trend is down in the most recent year, a significant backlog still exists.

This supports the consequences of Laney's observations in 2001.

3.10 Conclusions

In this chapter, we took an historical perspective on forensics in the analogue world to establish that digital forensics has inherited some expectations of process. When translated into the digital world, some of these processes work but some are found to have an uncomfortable fit. Until about 2007, this disjoint was tolerable but the rapid increase in data needing forensic analysis has pushed the current framework until it is close to failure. This is manifest in the backlog currently experienced by most facilities engaged in processing digital data.

In the next chapter we will review the possible solutions and choose a candidate likely to deliver the greatest benefit.

4 The Technical Considerations

In this section, we cover the technical factors that have brought about the disruptive changes over the last 10 years. Surprisingly, what most people perceive as universal improvements in technology, have caused problems for the digital forensic investigator.

We have already seen that there has been an increase in data generation in the last 10 years. In this chapter, we start by describing the architectures commonly found in forensic processing and explain why they are faltering. We then describe several solutions and choose distributed processing as being the one with most potential.

We then present our findings on distributed processing and that distributed processing, too, has had problems in dealing with very large amounts of data. We introduce Map/Reduce and the Hadoop file system and explain why we cannot simply adopt this as our platform. We look further afield to gather inspiration from other examples and identify the file system as being the component at the heart of a distributed system most likely to provide the controlled environment to satisfy our needs.

We then extend our boundaries of our solution to include imaging and a prioritisation system to deliver evidence to processing more promptly.

Finally, we draw together the subjects covered in chapter 4 and draw some conclusions that will be used to form our design criteria.

4.1 The Current Status of Equipment used for Analysis

Here we review the equipment commonly used within digital forensics. There may be more powerful processing facilities available to the higher security agencies but for the local or regional digital forensic facilities, the equipment is surprisingly familiar. Although this equipment is powerful, there are a number of mismatches between the component parts of the system. This means that they are not always fully utilising the processing power that is actually available.

4.1.1 ‘Conventional’ Processing Architectures

During the period 1990-2005 digital forensic processing, understandably, adopted existing system architectures as the platform for analysis software.

4.1.1.1 Stand Alone PCs

The simplest form found in a digital forensic laboratory is a single PC, probably with a multicore processor and local hard disk. Overall data transfer rate, I/O, is a combination of read speed from the media and data transfer rate through the interface. A typical hard disk can sustain reading data at about 80 MB/s while a Solid Sate drive can sustain reading data at about 450MB/s. The SATA III interface can transfer data at about 600MB/s; connecting a hard disk via a SATA III interface does not make the read speed faster from 80 MB/s. In our experience, programs like FTK 4 can process data at about 2MB/s per core on an i7 processor with typical options set. In digital forensics, the system becomes constrained by the data processing rate of the processor.

This is confirmed by Roussev's (2013) 'File Metadata Extraction' test. The consequence is that with even the biggest, and most expensive, processors, i.e. Dual Xeon with 16 cores in each socket only about 64 MB of data can currently be processed every second.

4.1.1.2 Networked PCs

One major disadvantage of the single host solution is that one forensic case is restricted to one host at any one time. If a host is in use, it blocks access to any other data stored on that host. The most obvious solution is to adopt a network of connected hosts for processing and have a central file store of forensic images. When this is done, the system bottleneck moves to the network connection.

Even when gigabit networking is used, the data transfer speed over the network becomes the constraining factor. Gigabit network data transfer is well balanced with hard disk transfer rates when it is used in typical commercial applications but as Gigabit Ethernet transfers data at just under 100MBytes/s it is enough to support just to provide data to 50 cores at 2MB/s per core. Gigabit Ethernet connections can be grouped together to improve the data transfer rate. Often referred to as channel bonding or aggregation, this allows up to 16 Gigabit channels to be joined in one virtual connection. This is a significant improvement but now the data transfer rate off the storage device becomes the bottleneck as it can only supply data at 100 or 450 MBytes/s depending on whether it is a hard disk or Solid Sate device.

4.1.2 The Processor/Storage/Data Transfer Gap

The previous section introduced the idea of a balance between the components in a computer system. Here we attempt to make sense of the current status of computer component performance and using common sense conclude that the current trend will continue for some time but at a slower rate of increase.

A computer system is a collection of sub-systems and components. These have developed at different rates. Table 9 contains an indicative guide to the relative capabilities of four important subsystems within a computer system.

	CPU	MIPS	Media Type	Disk Storage Gigabytes	Media Read Speed Mbytes/s	Disk Interface	Interface Speed Mbytes/s	Networking Mbits/s (Mbytes/s)
1995	80486DX4	70	Hard Disk	1	100	ATA-1	8	10 (1)
1996								
1997				2				
1998								
1999	Pentium III	2000		10		ATA-4	33	
2000						ATA-5	66	100 (10)
2001				30				
2002	Pentium IV	10000		40		ATA-6	100	
2003						SATA-I	150	1000 (100)
2004				128				
2005	Pentium E4	11000				SATA-II	300	
2006				300				
2007				750				
2008	i7, 4 core	82000	SSD		450	SATA-III	600	
2009								
2010				1000				
2011	i7 Extreme	177700		4000				
2012								
2013								

Table 9 - Comparison of the Relative Speeds and capacities of key computer components since 1995

PCs performance is a complex matter and there are many factors that dictate the ultimate processing power. There are, however, a number of key components.

- Processor clock speed;
- Number of processor cores;
- Storage media capacity;
- Storage media read speed;
- Storage media interface speed;
- Data bus speed.

4.1.2.1 Processor Speeds

In 1965 Gordon Moore (Moore 1965), then a director at Fairchild Semiconductor, speculated that by 1975 it would be possible to contain as many as 65,000 components on a single silicon chip. The annual rate of increase required to achieve this has varied over the years but, generally, held true until about 10 years ago when fundamental laws of physics prevented any further growth. It has been reworded to say “computer processing doubles in power every 24 months”. This observation has become part of computing folklore as “Moore’s Law”.

Moore's Law was true for single core processors. Beyond the development of single core processors, the subsequent speed improvement has largely been because of the adoption of Symmetric Multicore Processing (SMP) architectures with multiple cores on one chip. Essentially the 7i is faster than the i5 because the i7 has eight cores and the i5 only six. The i5, in turn, is faster than the i3 because the i3 has only four cores. Beyond this, costs rise disproportionately as more cores are added to the processor.

4.1.2.2 Media Capacity

From a typical hard disk of 1 Gigabyte capacity in 1995, hard disk have increased so much that, in 2014, 3 Terabytes drives are not uncommon with the largest currently being available at 6 Terabytes. This equates to an increase in media capacity of about 4000 fold in the last 20 years. Although data densities may be approaching a limit set by physics, it is likely that the increase in capacity will be continued by greater reliance on composite devices such as RAID arrays.

In PhysOrg.com, Mark Kryder (Kryder & Kim 2009) projected that if hard drives continue to progress at their current rate we can expect a 2 inch platter will be capable of storing more than 40 TB and cost about $40 in 2020. This is in line with Kryder's earlier prediction (Walter 2005) , in what has become known as 'Kryder's Law', that storage density will double more rapidly than Moore's Law. It seems that it is unlikely that that prediction will come to pass (The Register 2014) but the increase is still significant.

4.1.2.3 Media data Transfer Speeds

For most of the last 20 years, media data transfer speeds have largely been dependant on the mechanical nature of the components that make up the most commonly used storage device, a hard disk. Although small changes are being made, it is unlikely that there will be a significant increase in the read speed of hard drives from the current standard of about 100 Mbytes/s. The consequences of this on the time it takes to image 'the whole media' are linear. Typically, one terabyte takes 6 hours, so six terabytes, the largest drive currently available, takes about 24 hours. The use of light gases such as helium to fill the hard disk enclosure will increase operational speeds but not by much.

Because of the lack of reliance on mechanical components, Solid State devices exhibit much better read speeds, currently about 450 MB/sec. This is limited by the speed at which the circuits can switch within the devices.

4.1.2.4 Data interface Transfer Speeds

The last 20 years have seen a gradual increase from ATA-1 at eight Mbytes/s to the current top standard of SATA-III at 600 Mbytes/s. This is limited by either the switch speeds of the electronic component that make up the interface electronics or the encoding method such as the change from parallel to serial data transfer. Overall, this represents an increase of about 100 fold in 20 years.

4.1.2.5 Bus Speeds

This is similar to the data interface transfer speed; being limited by the physics of the electronic components. The bus speed is of little concern as it so fast beyond the other components that it has no constraining effect.

4.1.2.6 Other devices

USB memory sticks were available to the public shorty after they were patented in 1999 (Patent US6148354 Architecture for a universal serial bus-based PC flash disk Amir Ban et al".) with capacities of about 8Mbytes via a USB 1.1 connectivity of about 1MByte/s. They currently provide up to 512 Gigabytes of storage with USB III connectivity at 600Mbytes/se but are limited by the electronics within the memory to about 50 Mbytes/s read speed. DVD drives were introduced in the mid-1990s and has remained constant throughout this period. In all of these devices, generally, increases in capacity has outstripped increases in data transfer speed.

4.1.2.7 In graphical form

Figure 19 and Figure 20 show the gap in graphical form. In Figure 19, we see that disk capacity and processing power have increased by thousands of times in the last 20 years but network speeds and data transfer rates have hardly improved at all by comparison. Figure 20 shows this with a logarithmic scale and shows the clear division into two groups. Disk capacity and processor speed, in blue and green, have shown improvements noticeably better than media and network data transfer rates, in red and purple.

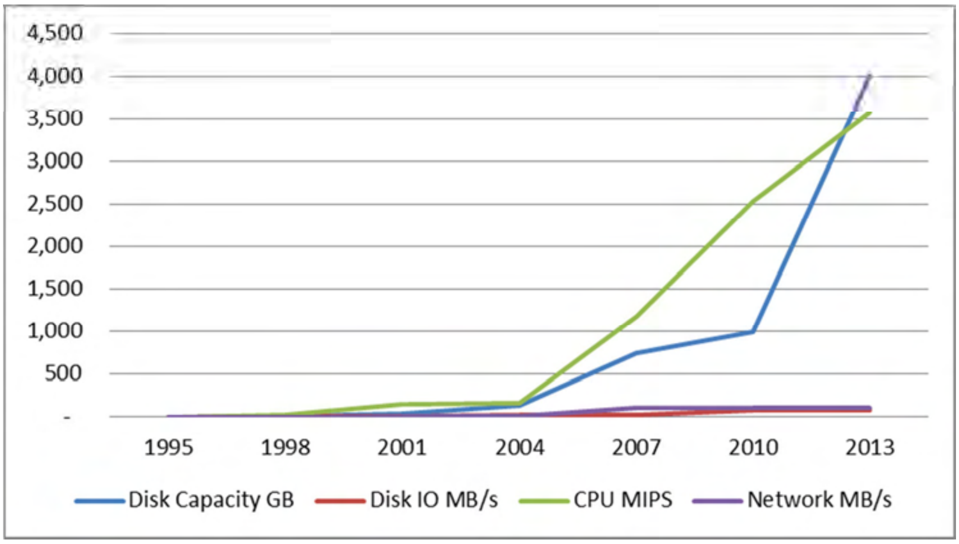


Figure 19 - The Change in power Ratios over 20 years

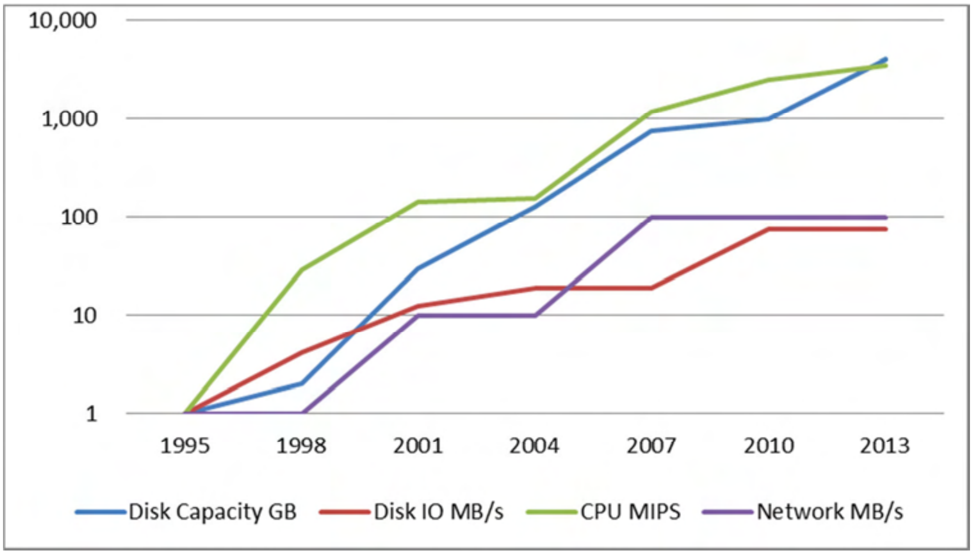


Figure 20 - The Change in power Ratios over 20 years (Logarithmic Scale)

Finally, Figure 21 (Interxion 2015) shows ‘Moore’s Law’ against ‘Kryder’s Law’ demonstrating the trend of media storage growth out-stripping processing Clock speed in Apple products between 1985 and 2005.

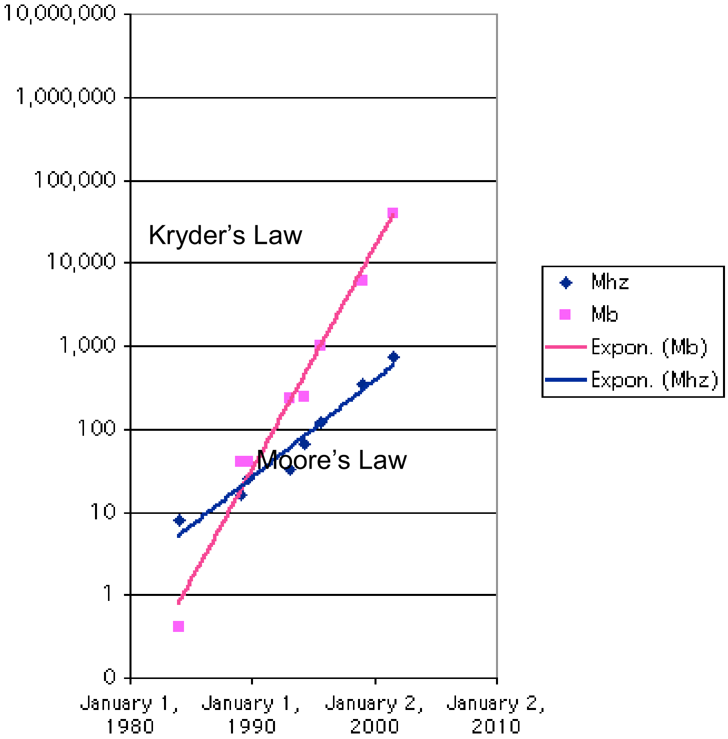


Figure 21 - Moore's Law v Kryder's Law (Interxion 2015)

Although there are breakthroughs in electronics and data encoding, both are out shadowed by the increase in media capacity. It seems that it unlikely that this differential will be reduced in the near future. As ordinary computer users are able to initiate actions that process data to satisfy their needs, like watching HD video, the computer industry will not develop and offer higher power data transfer at the lower prices consumers will be willing to pay.

There is an ever-increasing gap between quantity of data stored, and therefore needing to be processed, and our ability to process it in a timely manner caused by the relatively small increase in the ability to move the data around the system.

4.2 Various solutions

We can see from the figures of data capacity in 4.1.2.1 and data transfer rates in 4.1.2.3 that there is an ever-growing gap in our ability to process ‘all the data’. Here we consider a series of solutions, their likelihood of being a reality and their effectiveness.

In the early 2000s it was possible to ‘image the whole media’ in about an hour and ‘pre-process all the data’ in a couple of hours, ready for an investigator to analyse it. In 2014, it can take 24 hours to image a 4TB hard disk. It can then take another 24 hours to transfer it into the system and hundreds of hours to pre-process it. It is not feasible for a single investigator to analyse the entire contents in anything like a reasonable time.

Quick & Choo (2014) provide the best comprehensive review of all the literature associated with the issue of data volume in digital forensics. The focus of their research is data reduction but they widen this to include a review of Data Mining, Digital forensics As A Service, Distributed Parallel, Intelligence, Machine Learning, Triage and Visualisation.

In reviewing the relevant literature, we can classify solutions into five areas:

- Scaling UP in which we build bigger single hosts;
- Scaling OUT in which we build clusters of smaller hosts;
- Improving programming techniques with data-Mining, Machine Intelligence and sophisticated algorithms;
- Improving Management including by Triage and more people;
- Data reduction.

We now review each of these options and chose one that is most likely to yield the greatest advantage with the least effort in a reasonable amount of time. We need to find a solution, or blend of solutions, that will return us to the status of the early 2000s.

4.2.1 Scaling UP – higher specification single hosts

The most obvious solution seems to be, simply increase the power of the individual components.

4.2.1.1 Multi-Core Processors and Multiple Processors

This is largely what has been adopted over the last 20 years and could be summarised as the ‘wait until Intel solve the problem’ approach. As we saw in section 4.1.2.1, processor speeds have increased but we would need a many fold increase in power to provide the advantage to place us ahead of the demand. Exotic solutions such as quantum computing and processors based on light rather than electricity are, in all honesty, some time away from being available as commercial products. Multi-core and multiple processors solutions are disproportionately expensive as they increase in complexity. As the register (The Register 2012) reports an SGI (SGI 2012) 128-core with Xeon E5-4600s and 1TB RAM will cost $98,000. If the 128-cores were in 16 x i7 packages, each with 64GB RAM, the system cost would be, perhaps, $25,000.

4.2.1.2 GPUs

The use of Graphical Processor Units (GPUs) has been proposed (Marziale et al. 2007) but has its limitations. GPUs were originally intended to provide a processing platform for very fast calculations associated with rendering the 3D graphical images associated with visualisation and games. Their typical application is to load a dataset of information about objects in the virtual world into the GPU RAM and then process it for display. GPUs are particularly optimised for the rotation and translations calculations involved in plotting 3D worlds into two dimensions on a display. In addition, they now have optimised circuitry to decode video streams into high definition presentation. These devices are ideal for brute-force password cracking but would not provide much uplift when attempting to mine data from a large set of emails as in the Enron email set, as the data transfer rate into the GPU RAM is disproportionately slower than the processing. Another disadvantage is that they require special programming skills to exploit their architecture with programming environments like Compute Unified Device Architecture (CUDA); existing software will not run without extensive modification.

4.2.1.3 Improve Media and data transfer I/O Rates

This very much related to the previous section. However, there are a number of factors that limit development of storage system solutions. At the current time, the most cost effective storage solution is still a hard disk. Solid State Devices are about 10 times as expensive per GB. Although SSDs will be increasingly used to extend battery life in portable computing, it seems likely that hard disks will be the media of choice for mass storage for some time. Hard disks are mechanical devices where the platter spins at a specific rate. Standard disks spin at 5,400 rpm, there are high performance disk that spin at 15,000 rpm. SATA 3.2 is 1.6GBytes/s uses PCI Express. Intel (Intel 2014) currently produce an 800GB DC P3700 SSD drive with a PCI 3.0 x4 interface but at about $2,500 it is $3.1/GByte of storage. This compares with a four Terabyte hard disk that is about $35 and so is about $.0.01/GByte.

4.2.1.4 Optimising Existing Systems with RAID etc

This is, in a way, an extension of the previous solution. If individual data transfer rates cannot be improved, then by bonding channels together the overall rate can be improved. In storage media, this can be seen in RAID arrays and in networking this can be seen in aggregation or bonding of Ethernet wires. These can give an uplift of, typically, fivefold but are unable to go further.

4.2.2 Improving programming and analytical techniques

Improving processing by data mining, data reduction and machine intelligence is highly desirable but does require progress in many areas. Each problem needs a winning solution. There are many individual problems and so we need many individual solutions.

4.2.3 Improving Case Management and Triage

These usually involve discarding evidence or whole cases that are deemed of lower value or consequence or discarding parts of the investigative process. While it does provide some reduction in the current figures, it does raise the question of compromising the integrity of the investigation. Parsonage (2009) reports that the Nottinghamshire Police have the agreement of the Crown Prosecution Service in cases in a limited set of circumstances. This is only agreed where the case is one of simple possession of indecent images, the images found have been put to the suspect and they have made an unequivocal admission as to culpability. Parsonage reports a reduction of 254 to 139 (-42%) outstanding cases and a reduction the backlog down from 12 months to 7 months (-45%).

Although these results are impressive, it is difficult to see further reductions won at anything like the same effort. Further reductions are likely to meet serious obstruction with questions about the rigour of the investigation.

As Pollitt (2013) argues, triage approaches are certainly open to criticism.

4.2.4 Scaling OUT - Distributed processing in a Cluster

As we have seen, the exponential cost of improving the performance on one box was noticed in the mid-1960s. We will explain this further in 4.3.1 but this lead to another strand of development, scaling out. In this model, we connect together a cluster of cheaper components, using Mass Market Commercial off the Shelf components (M^2COTS), and share work out across the cluster. This is not as powerful as a single box as it requires data transfer between nodes within the cluster at the lower speeds associated with local and wide area networks but it is far more cost effective. The key to success with distributed processing is data dispersal and storage. Here the emphasis is on value for money rather than processing power at any cost.

4.2.5 The need to improve the process

The combination of the earlier process models and the linear nature of forensic imaging has become such a solid founding principle for a new subject that we have been very reluctant to move on from these sequential, linear processes to models that are more concurrent and so are able to exploit new computer system architectures. However, the success of distributed processing hangs largely on the gains from concurrent processing taking place across the nodes of the cluster.

To clarify this, we can return to Pollitt's original but modify it slightly it to five blocks, Figure 22. Each block represents a collection of activities and could represent a considerable resource commitment when examining large media.

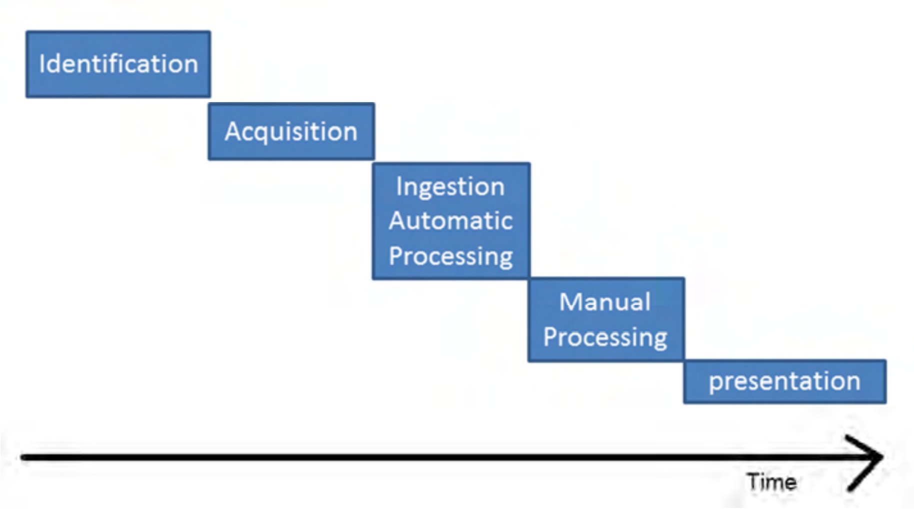


Figure 22 - Linear Processing

A real solution would be to move to a much more optimized process, such as that in Figure 23, where each successive stage starts as soon as it possibly can after the commencement of the previous stage.

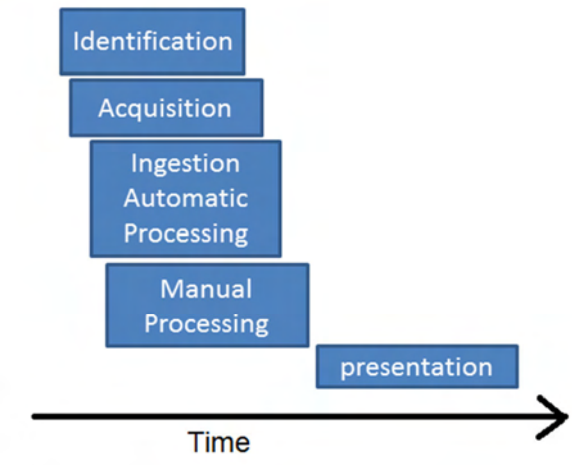


Figure 23 – More optimised processing

Figure 23 shows a series of processes that are nearly completely concurrent given there is a slight delay on each subsequent stage as administrative tasks need to be completed to initiate the task but each task is largely exclusive of other tasks.

Where multiple tasks are capable of running mutually exclusively of each other's processing it is said that they are "embarrassingly easy to parallelise" by the High Performance

Computing community. Each task is never dependant on another task's results or progress. In most cases, forensic data analysis is "embarrassingly easy to parallelise" in that the processing, for example of one JPG to extract EXIF information is not dependant on the extraction of EXIF information from another. Although we often need to associate resulting data between files, the processing of raw data itself rarely needs to cross-reference between files. JPGs, TIFs, PDFs and PSTs are all examples of self-contained files without reference to external data. The extraction of EXIF information from one photograph is not dependant on the extraction of data from any other file.

Accounting systems and databases often require reference to data outside of a single file but within the files that constitute a 'set' of data files they, too, are "embarrassingly easy to parallelise".

4.2.6 Why is distributed processing the best solution in forensic processing

When we consider the previous solutions there are advantages and disadvantages in all of them. In section 1.3.5, when we set out the scope of this project, we set ourselves a design constraint that the solution must be within the budget of a regional forensic crime facility and subsequently set ourselves a hypothetical budget of about £30,000 at 2014 prices. This effectively rules out the scaling up solution as being too expensive. In addition, we might see speed improvements double every year but it would be preferable to have a solution that can be scaled in much shorter time scales.

Clever programming is clearly desirable but has to be applied to every problem individually. There is constant work in a variety of areas for example "On the database lookup problem of approximate matching" (Breitinger 2014) but it is piece meal and has limitations because of the need for inventive thinking. It may be that many solutions are already at or near their optimum and if improvements were made it might be of a small increment.

Management triage is currently leading the way in reducing the backlog but there is a risk in triage that the best triage assessment may be wrong and vital information lost in the cut.

Scaling out, with Distributed processing, has many advantages. It uses existing technology but can utilise new developments in technology like SSDs and GPUs. Because Distributed processing is based on the idea of combining large numbers of standard devices running standard operating system, it has an advantage of being usable 'off the shelf' and has the prospect of yielding some useful results in the short term. In addition there is a much lower entry cost in its development and it is scalable with regards the size of the problem and the budget available to solve it. There is no necessity to purchase expensive new devices as in the SGI UV 1000 described in section 4.2.1.1

4.2.7 Distributed Processing

Within digital forensics, the implementation of distributed processing to improve the processing of data was first proposed in 2004 with Roussev's publication (Roussev & Richard III 2004) and the DELV system. In it, Roussev asked a key question:

> *"In summary, the case for going distributed in forensic analysis is not fundamentally different than in any other application domain where high performance is needed. The obvious question is: should we try to use generic distributed frameworks (GDF) and adopt them for our purposes, or are we better off developing a more specialized solution? Our own survey of available GDFs (in the Related Work section) leads us to believe that a specialized solution is a better choice for a number of reasons:…."*

We shall see that a decade later our own assessment came to the same conclusion.

Roussev forms a list of requirements for a Distributed Digital Forensic Toolkit:

- Scalability;
- Platform-independence;
- Lightweight
  - Efficiency
  - Easy administration;
- Interactivity;
- Extensibility;
- Robustness.

Roussev does not mention information assurance within the design of DELV. DELV will be the subject of more detailed, technical analysis in section 4.3.2 but at this stage its seems that Roussev was relying on the assurance of basing the system on forensic images stored on a central file store, as we identified in section 3.6.4.

The unpublished MSc work (Pringle 2004) has the first building blocks of this research. In it, a design for a forensic tool with Client-Server architecture was proposed and built as a prototype. The evidence was stored on remote server in the form of an image, with a client program that presented a user interface, issued commands to the server in an XML format. After the server had parsed the request and processed it, the results, also in XML format, were sent back to the client to be displayed. Within the design is a very simple error checking function that guarded against data corruption between the processing server and the visualisation client.

The next reference to the use of Distributed Computing to enhance Digital Forensic processing was by Golden Richard and Vassil Roussev in "Next Generation Digital

Forensics" in 2006. (Richard & Roussev 2006). This was published about 2 years after their DELV system. In their book chapter, they firstly reassert that

> *"… massive increases in storage capacity for target devices are on the horizon. The traditional approach of utilizing a single workstation to perform a digital forensics investigation against a single evidence source (e.g., a hard drive) will become completely intractable as storage capacities of hundreds of gigabytes or terabytes are seen more often in the lab."*

And that

> *"Generally, there are two possible approaches to improve machine scalability—improve the efficiency of the algorithms and their implementations to get more from the current hardware platforms or enable the use of more machine resources in a distributed fashion. These two approaches are to a great extent complimentary; however, the former is likely to yield only incremental improvements in performance, whereas the latter has the potential to bridge the hardware performance gaps discussed earlier".*

It is clear that they see distributed processing as part of a multifaceted approach to solving the processing problem. Further, refining their view, they note that

> *"Maximizing CPU utilization is a bit more complicated. One approach is to scatter the files of a particular type evenly across the processing nodes. The rationale is that whenever an operation is issued, for example, a regular expression search, all nodes will have a similar amount of work to complete and, therefore, CPU utilization will be maximized. However, more sophisticated processing that attempts to correlate different objects (such as the image classification technique discussed later) may be hampered by this file distribution pattern, increasing the need for network communication. In this case, concentrating the files in fewer nodes and crafting a suitable communication pattern may yield better results".*

This refers to the classical load-balancing problems that have always been present in the design criteria of computer clusters.

In their conclusions they summarise with

> *"The technical challenges facing next generation digital forensics tools are dominated by issues of scale. Current single-CPU systems are quickly*

> *approaching a point where their poor performance will make them unusable, due to a fundamental imbalance between the resources needed to process the target and the resources available on a single forensics workstation."*

Pringle and Sutherland (2008) built a Globus 4 GRID cluster (Foster 2005) with the intension of testing its suitability as a platform for digital forensics and to test Roussev's claim in "Breaking the Performance Wall" that existing cluster systems were heavy weight and difficult to use.

Creating a Globus GRID from scratch was difficult but this was made considerably more easy to setup using the Instant-Grid LiveCD (Instant-Grid 2013).

Efforts were made to mimic the DELV architecture and the processing tasks in Roussev's 2004 prototype as closely as possible. In mimicking DELV, data was stored as an image on a central file store and was distributed out to the processing nodes. Batch programs were written to control data transfer and initiation of the programs running on hosts within the cluster. When configure with no authentication processing times were comparable with the reported by Roussev with DELV.

GRID has the option to use strong security based upon Kerberos authentication. When this was used, task initiation became a significant delay in the system. In addition, as with DELV, when the data was eventually sent and stored on each processing-node, processing became much more efficient and immediate. This was even more apparent when the data was cached in RAM on the processing node.

This largely disproved Roussev's assertion that existing cluster systems were too bloated to be used. However, there are significant differences in the nature of processing needed to process different types of data efficiently. Digital forensics processing could be one of two very different problems. One extreme, in which a single very large file, a single forensic image of a multi-terabyte disk, needs to be processed or the other extreme of needing to process perhaps, millions of individual files that make up the original file system. There is a more detailed technical assessment of this in section 4.3.3.

The issues of using distributed processing for forensic data is addressed more comprehensively in 2009 by Daniel Ayers in "A second generation computer forensic analysis system" (Ayers 2009). Ayers states

> *"A set of requirements for second generation tools are proposed. A high-level design for a (work in progress) second generation computer forensic analysis system is presented".*

He has made no further publications about the proposed system but has filed three patent applications in New Zealand (NZ579120, NZ595049, NZ59052).

Ayers starts by observing and repeating Roussev's observation from 5 years earlier that

> *"The architecture of existing – first generation – computer forensic tools, including the widely used EnCase and FTK products, is rapidly becoming outdated. Tools are not keeping pace with increased complexity and data volumes of modern investigations"*

and then

> *"First generation computer forensic tools are struggling to keep pace with modern analysis workloads. Even when deployed on expensive high-end workstations with multiple processor cores, large amounts of memory and fast disk storage the ability of a single (even multi-threaded) application to quickly process evidence data is constrained."*

Ayers agrees with Richard and Roussev that

> *"The greatest improvement in I/O throughput will be achieved through efficient design of on-disk data storage and use of parallel processing."*

Ayers then extends Richard and Roussev's ideas by observing that

> *"The use of parallel processing to provide additional processing capacity is an important advance in computer forensic tools. However, this addresses only one of the significant limitations of first generation tools, and for that reason I describe such tools as ''Generation 1.5''. Other issues such as tool reliability, auditability, data abstraction, efficient data storage and repeatability of results must also be addressed if computer forensic tools are to truly move into a ''second generation''.*

Ayers summarises the idea of a series of metrics by which it would be possible to judge the efficiency and performance of computer forensic tools under the headings of:

- Absolute Speed: measured by the time elapsed from start to finish
- Relative Speed: a ratio between the read speed of the storage media and the processing speed
- Accuracy: the proportion of results returned that are correct
- Completeness: the proportion of evidence found
- Reliability: that the tool does not crash and recovers from errors

- Auditability: that it's actions can be verified

He draws attention to a goal objective within forensics reminding us that

> *"Analysts must realise that the objective of a computer forensic analysis is not to locate relevant files but to identify relevant evidence. The analyst's ability to recognise and analyse certain types of relevant evidence can be improved if that evidence is presented at a higher level of abstraction than computer files."*

Ayers identifies the apparent conflict between the established practice of a forensic image and the need to store data in a form more suited to distributed processing.

> *"While the raw image of an evidence item is considered to be the ''original'' version of the evidence and must be retained, it is often not the optimal format to store evidence in for later processing,"*

Ayers does not make a clear distinction between application software, the operating environment and any middleware. He refers to 'the tool' as a singular entity throughout his paper. "The Tool" tackles everything from the processing of data to its storage, distribution and work scheduling.

He defines the term "second generation computer forensics tool" by defining a criterion that the tool may or should meet:

On Parallel Processing. Ayers believes 'the tool'

> *"… [It] must be able to use the computational resources of many separate processors (i.e. processors that do not share main memory or I/O bus bandwidth) so as to be capable of improved absolute and relative speed. The tool must be able to process data volumes that exceed the aggregate RAM of all processors by at least an order of magnitude. An automated method for distributing evidence data around processors and collecting results must be provided. The tool should be able to use processors with different operating systems and processor architectures. The tool may make use of distant ''grid computing'' resources providing that evidential integrity and confidentiality is maintained.*

On data storage and I/O bandwidth, he believes "the tool"

> *"… [It] must support a fault tolerant, high performance and scalable data storage medium so that analysts can implement a data storage solution to meet arbitrary capacity and throughput criteria. The data storage medium*

> *should support a range of computer architectures and operating systems. The tool should store evidential data in a form that ensures it may be efficiently accessed.*

On software tool assurance, data assurance and auditability, he says:

> *"Source code for forensic analysis functions should be available for independent review by a qualified third party. Ideally, this requirement would be met by making source code publicly available, although in the case of closed-source tools a detailed source code review and acceptance test by an independent auditor could be substituted. The tool must be able to generate a detailed log of all evidence parsing, analysis and searching activities. The tool must maintain an audit trail record of the actions of each analyst. The tool must be able to record and display to the analyst, in a convenient form, details of all computations undertaken to produce any result together with details of any assumptions used in those computations and any other factor (such as configuration data or information provided by the operating system) capable of influencing the outcome of a particular computation. The tool must clearly identify which results are merely displayed and which are the result of computations or are influenced by configuration data or information provided by the operating system. It must be possible to trace each result and/or fragment of evidence back to the original raw data in an evidence file. The tool must ensure that the integrity of evidential data is provably maintained."*

Ayers was the first to consider the broader issues of the application of a distributed processing approach to processing of data for forensics and notes that

> *"Grid computing is so different to the normal way of conducting computer forensic analysis that it may be difficult to convince a Court that results obtained using that technique are reliable and therefore admissible."*

Ayers concludes with

> *"Digital forensics investigators have access to a wide variety of tools, both commercial and open source, which assist in the preservation and analysis of digital evidence. Unfortunately, most current digital forensics tools fall short in several ways. First, they are unable to cope with the ever-increasing storage capacity of target devices. As these storage capacities creep into hundreds of gigabytes or terabytes, the traditional approach of utilizing a single workstation to perform a digital forensics investigation*

> *against a single evidence source (e.g., a hard drive) will become completely intractable. Further, huge targets will require more sophisticated analysis techniques, such as automated categorization of images. We believe that the next generation of digital forensics tools will employ high-performance computing, more sophisticated evidence discovery and analysis techniques, and better collaborative functions to allow digital forensics investigators to perform investigations much more efficiently than they do today."*

Garfinkel (Garfinkel 2012) discusses the choice of platform and programming languages when writing software for digital forensics. He discusses the choice of C, C++ and Java as development language and the implications of speed advantage of multi-threaded programming. However, he does acknowledge that much software in digital forensics is written by knowledgeable non-programmers for whom the complexities of these more advanced process models prohibit the development of high performance software needed to return results in a reasonable time when processing large volumes of data.

On the specific matter of parallelism he reports that his results are mixed with little or no-success on GPU implementations but staggeringly successful results on making bulk-extractor multi-threaded.

Roussev et al. (2013) consider the idea of a time-constraint, or real-time limit, in the investigative process. Acknowledging that time is often an imperative in investigations they approach the problem by considering how much processing can be done in a set time. They observe that within a typical system used for forensic analysis, individual storage media can now supply data fast enough to occupy about 120-200 cores in typical i7 machines. They propose an architecture shown in Figure 24.

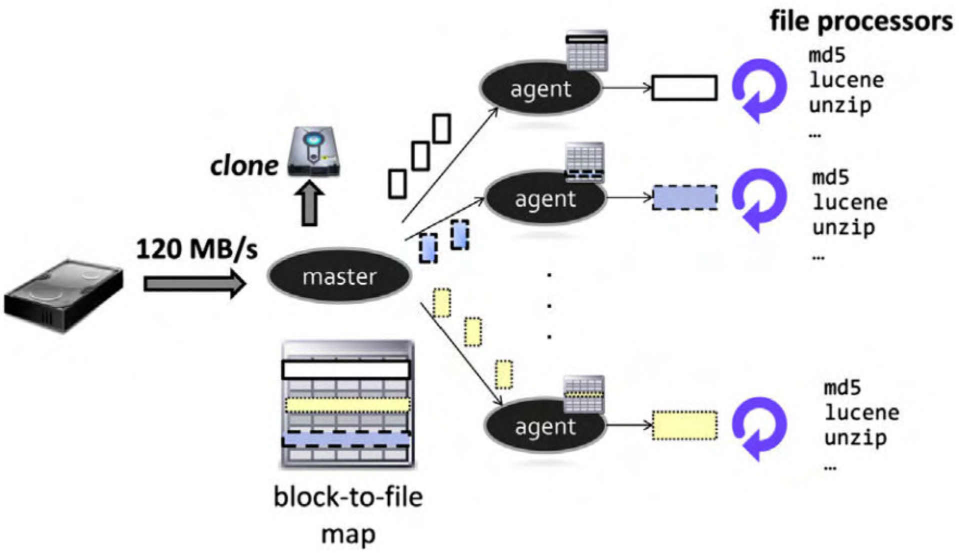


Figure 24 - Latency-optimised target acquisition architecture (Roussev et al. 2013).

This is based on a new paradigm in imaging, LOTA, where firstly the file meta-data is read, including the clusters occupied by the data, and as the media is imaged, cluster-by-cluster, completed files can be bags and sent off for processing.

> *"The rationale of the system is simple d before we start cloning a target, we parse its filesystem metadata to build an inverse map of data blocks to files. After that, we start reading the disk blocks sequentially from beginning to end, and use the map to reconstruct on the fly the files whose contents have been acquired. Once a file is complete, it is made available through the regular filesystem interface to file processing tools. Disk cloning proceeds in parallel."*

LOTA is examined in detail and compared with other imaging techniques in section 4.6.1.

As media sizes started to increase significantly in the first part of the last decade, it started to become clear that a unit of storage capable of storing multiple whole devices or data items smaller than a whole media image was needed. Two new forensically sound storage formats were introduced in 2006.

4.3 Distributed processing

Having chosen distributed processing as being the approach that will most likely yield greatest results, it is now time to review the history of distributed processing and consider some of the existing system as to their suitability for our purpose. There are many distributed system but there are only four that can be said to be in common use and display fundamentally different characteristics in their operation. In this review we will see that the most recent addition, Hadoop, introduces a new concept in distributed processing, that of data locality. We find that moving data to be processes is actually a counterproductive activity and should probably be avoided. Distributed processing has been proposed within digital forensics and subsequently we review, in some detail, Vassil Roussev's DELV system. We also review one of the most popular and trusted systems, AccessData's FTK, which is promoted as a distributed processing system but find that this is limited by the need to transfer data from a central store of forensic images out to processing nodes. The very activity that Hadoop is designed to avoid.

Programming in a distributed environment often means significant changes to existing programs and so we present the fundamental characteristics of programs that need to be considered when moving to such and environment.

Finally we present the Open Architecture Information Systems Standard which was encountered after the design for FCluster was completed. It contains important characteristics and thankfully, FCluster does seem to conform to its specification.

4.3.1 The Historic background of Distributed Processing

Electronic computing has its origins in the Second World War. At the time, processors were built from components like valves on circuit boards. In the late 1950 Jack Kilby proposed the idea of an integrated circuit and in 2000 won the Nobel prize in Physics for his work (Texas Instruments 2014). At this time the idea of more computing power meant more circuitry and so it is, perhaps, no surprise that the first moves to high performance computing were taken by attaching two processors together on the same board with 'simultaneous' access to shared memory (Wikipedia 2014a; Wikipedia 2014b). Very quickly, the number of attached processors increased. The key issue at this time was how to connect processors to each other, memory and I/O. There are a number of approaches. Symmetric Multiprocessing machines (SMP) arranged multiple processors, each executing its own program, but having access a shared memory. In a Multiple Instruction, Multiple Data (MIMD) processors are arranged to respond to a stream of commands that formed a program, each instruction can be sent to any available processor. Generally, in this model, each processor has its own local memory but they can share a central memory. The MIMD architecture suffers from scalability problems because interconnection becomes complex after 32 processors.

There are many variations on this multi-processor, shared memory, architectures but they share one common factor, and they are all expensive.

It is generally accepted that programming to exploit a tightly coupled parallel architecture efficiently is disproportionally more complex than single thread programming. The interrelatedness of the tasks adds a complexity of synchronisation that is generally not present in single thread programming. Eventually this may result in a state of a 'law of diminishing returns' where excessive effort is required to yield a minor return.

It is not surprising that there was a growing need for a more affordable and accessible form of HPC. This was satisfied in the mid-1980s. In his PhD thesis "The Study of Load Balancing Algorithms for Decentralised Distributed Processing", Miron Livny (Livny 1983) considered the theoretical aspects of the efficient co-ordination of independent host computers linked in a cluster. This formed the foundation for the first serious management system for distributed processing – Condor.

4.3.1.1 HTCondor

In "Distributed Computing in Practice: The Condor Experience" (Thain et al. 2005) we find a comprehensive review of the origins, design criteria and influences within the theory and practical implementation of the system by some of the original Condor project team at the University of Wisconsin-Madison. The project was renamed to HTCondor in 2012 to avoid confusion with another commercial product.

HTCondor is a high-throughput distributed batch system that utilises spare capacity of existing machines. Programs can be run on any suitable host within an HTCondor cluster without modification to the original program. If the program can be modified, implying that the source code is available, extra libraries can provide the enhanced abilities to ‘checkpoint’ the progress of the task and, as a consequence, provide the ability to suspend the execution of that task on one host and allocate it to another for further progress.

HTCondor highlights a distinction between tightly and loosely coupled problems. HTCondor is ideal for handling tasks that are said to be “embarrassingly easy to parallelise”. Each task should be capable of running as a ‘closed system’ without interaction to the user or with another running task. HTCondor recently introduced a facility to facilitate user interaction but it is most efficient when running in its batch model.

From their introduction Thain et al. observe, “Scientific interests began to recognize that coupled commodity machines were significantly less expensive than supercomputers of equivalent power”

It was accepted that although this architecture would have a higher management overhead, it would still deliver an acceptably high power to cost ratio.

The HTCondor architecture is in stark contrast to the closely coupled architecture of SMP and MIMD. On the one side was very loosely coupled, embarrassingly easy parallelisation and on the other tightly coupled shared memory processing. In 1993, a new paradigm was established to bridge the gap between tight and loose coupling.

4.3.1.2 Message Passing Interface

The Message Passing Interface (MPI) (Message passing Interface (MPI) 2014) defined the semantics and syntax of a core of library routines that would extend C and FORTRAN to enable the exchange of information in the form of messages between machines in a cluster. It has been extended to most popular languages. In this architecture hosts within a cluster, for the most part, operate individually and either pass or receive information in the form of messages. Message Passing has now come to characterise ‘Distributed Computing’. For example, ‘Monte Carlo simulations’, where random numbers are generated to test a hypothesis, can be run on separate hosts. Interim results can be passed between hosts as the correct result is approached. MPI introduces the principle of memory locality where processing takes place within memory directly addressable to the processor host. These clusters have often become known, collectively, as Beowulf clusters after one of the first that was developed at NASA in 1994. There is no strict definition of the characteristics of a Beowulf cluster but it is generally expected to consist of a local cluster of ‘normal’ commodity grade computers that normally run a UNIX like operating system and employing message passing as a control mechanism. Several software architectures have been developed to operate on Beowulf clusters. These systems did not replace the operating systems in the host computers; instead, it forms a layer between the user and the host operating system, a “Middleware” layer.

According to Enslow (Enslow 1978) there are 5 characteristics of Distributed Systems. Whatever the system is proposed it should satisfy these criteria.

- Multiplicity of Resources
  Distributed system have more than one, but probably many, hosts;

- Component Interconnection
  Components (hosts) are connected by a network system and software servers by logical pipes to transfer messages;

- Unit of Control
  There are programs running to control the cluster behaviour by allocating tasks, scheduling processing and clearing up after processing;

- System Transparency
  The user should not be aware of where or how the cluster achieves its ends. The control software should deal with this;

- Component Autonomy
  The system design should assume the certainty of failure and deal with it without the knowledge of the user. Ideally, the failure of one component should not affect any other.

Coulouris et al. (2012) adds “A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages”. This often manifests itself in the form of an additional software system that sits between the host operating system and the user’s programs. Not surprisingly, it is called Middleware. There are many examples of middleware in distributed processing; each design is intended to be optimised for a specific type of task. A brief examination of one commonly used system will highlight some of these characteristics.

HTCondor has grown out of the desire to utilise fully whatever processing resources are available. The key here is efficiency, embodied in the concept of load balancing. HTCondor has its origins in the PhD thesis of Miron Livny (Livny 1983). HT – High Throughput. From the HTCondor v7 manual

> *“A growing community is not concerned about operations per second, but operations per month or per year. Their problems are of a much larger scale. They are more interested in how many jobs they can complete over a long period of time instead of how fast an individual job can complete.*
>
> *The key to HTC is to efficiently harness the use of all available resources.”*

Livny believes that the salient characteristic of Distributed Systems is the multiplicity and autonomy of its resources. HTCondor runs concurrently on non-dedicated computers connected by a network. They do not have to be on the same sub-network i.e. a local cluster. Jobs are allocated to hosts based on a ‘Class Ad’ service that reports on the capabilities of the candidate machine and the current load state of the candidate. When a user submits a job to Condor, the system finds a suitable candidate host and sends both programs and data files to the machine.

Each host in an HTCondor cluster needs to be identified and its capabilities collated to be used as a ClassAd to enable the selection of a suitable host.

```
Machine = "turunmaa.cs.wisc.edu"
FileSystemDomain = "cs.wisc.edu"
Name = "turunmaa.cs.wisc.edu"
CondorPlatform = "$CondorPlatform: x86_rhap_5 $"
Cpus = 1
CondorVersion = "$CondorVersion: 7.6.3 Aug 18 2011 BuildID: 361356 $"
Requirements = ( START ) && ( IsValidCheckpointPlatform )
Memory = 1897
OpSys = "LINUX"
Arch = "INTEL"
Mips = 2634
Activity = "Idle"
TargetType = "Job"
CheckpointPlatform = "LINUX INTEL 2.6.x normal 0x40000000"
Disk = 92309744
VirtualMemory = 2069476
```

Figure 25 – An example of an HTCondor Classified Ad file

Jobs are defined in a text control file.

```
executable = mathematica
universe = vanilla
input = test.data
output = loop.out
error = loop.error
log = loop.log
request_memory = 1 GB
requirements = OpSys == "LINUX" && Arch =="INTEL"
rank = Memory >= 64
image_size = 28000
initialdir = run_1
queue
```

Figure 26 - HTCondor Job Definition

HTCondor itself selects an appropriate host based on the criteria described in the Job description being matched to a specification presented in a Classified Ad. The user plays no part in the allocation of a job to a host.

Condor assumes that the whole job, including data, will be transferred to the processing host once it has been allocated to a host. It then sends the results data file back to a location specified in the job file. If no specific source hosts are specified, it assumed the local host, from which the job was initiated is both the source of the data and the destination of the results.

4.3.1.3 BOINC, Distributed Community processing

BOINC (The University of California 2014) is a ‘community distributed processing system’ developed over the last decade. BOINC’s basic architecture is of a client-server model with the server, or cluster of servers, providing job management, allocation, distribution and finally collation of results and clients who do the actual processing. There are some features of BOINC that are of significant interest.

- The BOINC client collects benchmarking data about its host platform and returns this to the server. BOINC sends tasks triplets to clients running on comparable hosts;
- BOINC is designed to ‘survive’ unreliable or links which may disconnect. BOINC pre-loads clients with work packages before the preceding packages have finished. Results are stored and new tasks are started with or without a network connection. Data transfer resumes when a network connection to the server is available;
- BOINC projects, SetiAtHome for example, typically distribute more than one, usually three copies of a work package and compare the results when they are returned using what is termed a validator. The validator either completes a bitwise comparison of results or may employ some fuzzy logic in its validation process.

4.3.1.4 Distributed Processing with Hadoop

At about the same time as Roussev and Richard were designing DELV there were various groups addressing the problem of processing increasingly large quantities of data found across many fields of research. Of these, Hadoop has had the most influence in the last decade.

Internet search engines generate usage data in the form of massive log files that record such things as the text search strings and the selection of a link from the results list generated.

It was proposed that a new approach would successfully solve this data processing problem by a ‘Divide and Conquer’ processing model called MapReduce (Dean & Ghemawat 2004). The model is founded upon dividing very large files into regular blocks across many data nodes and processing each individually (White 2012).

These blocks are distributed across the cluster as a massive distributed file system. This can be thought of in the same way as clusters or sectors in a popular file system like FAT or NTFS when deployed on a RAID array, but on a huge scale.

NTFS clusters are typically 4K; HDFS's blocks are, by default, 64 MB. The choice of default cluster or block size is indicative of the overall size of data files expected to be stored and is a question of efficient use of disk space. Most files on a Windows XP system drive are less than 10k and so 4k is an efficient block size. Files stored within an HDFS file-system are expected to be many gigabytes, if not terabytes, and so 64 MB is efficient. We note that the HDFS block size can be increased to a maximum of about 4 GB, that is, a java long integer or 4,294,967,296 bytes. In the same way as data is distributed across different disks in a RAID array, the HDFS blocks are distributed across disks located in different hosts within a cluster or even across data centres. It is important to note that the whole file is divided into exactly 64 MB blocks regardless of any alignment to 'data record'. Words, sentences, URLs etc. are routinely cut at the end of the block and subsequently continued at the beginning of the next block.

When a program is submitted to Hadoop, it employs the MapReduce model within the 'tasktracker' control program to initiate processes that run on each DataNode and process data held locally in each discrete block on the DataNode. The tasktracker tries to avoid the 'expensive' act of moving the data to the processing host preferring processing to be done locally. The programs initiated by Tasktracker, for the most part, only process data held on local storage.

Each of these instances (and there will be many instances running at the same time) read data in units decided by the programmer. They may be of a fixed length or delimited by expected markers like spaces, commas or carriage returns.

Data is read from the first block from the start of that block assuming, quite rightly, this is the start of the data. When it comes to the final input of data from the block, which will almost certainly be incomplete, it retrieves the location of the next data block from the NameNode (the HDFS directory service) and requests data from the DataNode to be sent to complete the input block. By default, this is 16k. This is repeated, if and as necessary, until a suitable end of input is met. When set against a 64 MB block size, a 16kb transfer is minimal.

For the middle blocks, MapReduce ignores any data before the first occurrence of either a multiple of the fixed record length or an anticipated delimiter, whichever the programmer chose, as that data was processed with the data in the prior block. It then supplies data to the task instance by a simple streaming input method. The last input of data is completed using the same method as in the first block.

In the final block, it again discards data until the end of the first, incomplete record, and then reads data until the end of the file and, quite rightly, stops.

Figure 27 show a schematic of the data transfer within HDFS when reading a short list of words spread across 4 blocks that are held on 3 hosts (A, C and Z). The programmer would have chosen the space character (represented by the _ symbol) as an end of input marker.

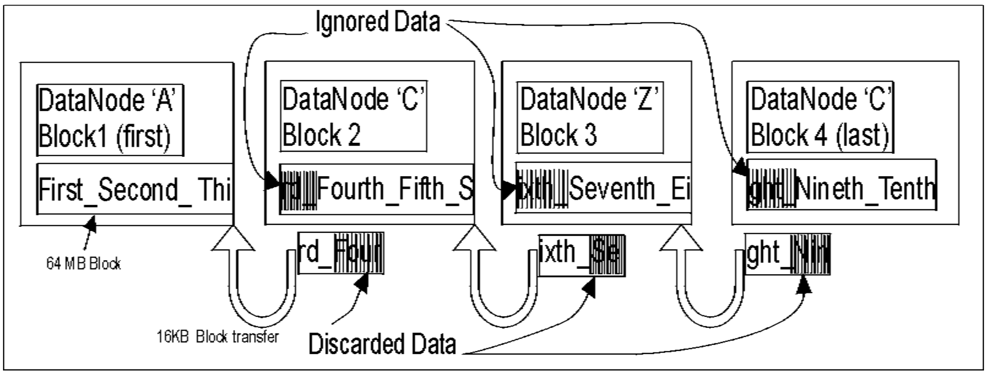


Figure 27 - Data transfer between blocks within HDFS

The Job Tacker issues an instruction to the Task Tracker on each of the three hosts. The Task tracker on DataNode C, which has two data blocks, initiates two tasks while the Task Trackers on hosts DataNodeA and DataNodeB initiate just one each- Figure 28.

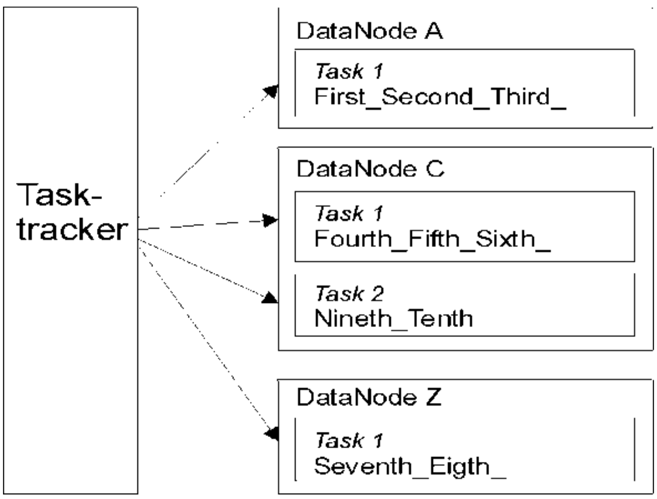


Figure 28 - Tasks allocated to Datanodes

Having read and processed each of the input units generated by "mapping", the resulting data is the returned to the initiating host to undergo post processes, "reducing", typically by, concatenation and perhaps sorting. Hadoop has a workflow system to enable this cycle to be repeated for further processing of data if required.

For the type of data collected in log files, this is very efficient and is nearly linear in its scalability. This is so successful it is now employed as a model in many different fields from monitoring stock prices or customer purchases to processing weather station data and, famously, the Large Hadron Collider. Anywhere where the data resembles log files.

However, this almost completely prevents efficient random access across a file. In our previous example, if a task, running on DataNode A, wants data that is at an off-set 'n' in the current input file, the task would have to send a request to the NameNode, which contains and controls the file-system meta-data, asking for the location of the block containing that data. Let us say it is on DataNode C. The task would then have to initiate a transfer of that data from DataNode C in order to process the data.

The only way to achieve random access within Apache Hadoop is to use the FUSE file system access and run the task on a single host, in which case all data must be transferred to the processing host. It is likely this would be worse than local processing of data. This is likely to be primary the reason HBase, built upon Hadoop, has no built in join (Apache 2014)

It should be clear that although these file systems can be used they are not of an optimum design for use in digital forensics.

We believe two groups are using Hadoop for forensics. The Dutch National Forensics Institute in Holland and Lightbox, in America, have developed systems that use Hadoop to process information previously extracted from the original forensic image.

4.3.1.5 The Significance of Moving Data

The movement of data off a central file server is a significant factor in determining the overall performance of the system and is a key limitation on scaling of the system and demonstrates the ultimate failing in this architecture.

If we say,

- V is the volume of data to process
- P is the rate at which a single core can process data (a broad average rate of ~2MB/s)
- N is the number of cores in a single host
- T is the rate at which data can be transferred between hosts (~ 70 MB/s)
- I is the rate at which data can be transferred from the storage media hard disk (~ 100 MB/s)

Host utilisation is $\frac{(N*P\ )}{T}$

On a single host, when $I > N * P$ then the single host is underutilised.

On a single host when $I < N * P$ then data transfer is worth considering.

But transferring data to another host is only worthwhile when $(P * N)_{host} > T$

If T is 70, P is 2 and N is 8 then one file server can support T/(P*N) hosts, 70/(2*8) = ~5 hosts.

It must be said that the current processing by FTK is basic. If we anticipate future developments in machine intelligence or data reduction it is highly like that the processing rate will reduce from the current average of 2MB/s to a fraction of that.

There is a complication concerning the size of files. Opening a file, transferring a block of data and closing a file each take a set amount of time. Therefore, data transfer can be said to be:

$$Topen + \ T\ block * blocks + T\ close$$

- Small files, perhaps under 20k, take a disproportionately long time to open and close when compared with the time it takes to transfer its data. Transferring 10,000 files of 10,000 bytes takes very much longer than one file of 10,000,000 bytes.

- Individual files can also be very large, as in the case of video files. As the number of blocks increases, the overhead of opening and closing the file becomes proportionately less in the overall time.

In this case, simple queuing theory can be used to reduce overall processing time by selecting a large file and transferring it to a suitable host either for processing and then continuing to process further, presumably smaller, files locally or by passing them out to a suitable host.

A well-balanced system should have just the right amount of data storage, with an interface that could support the available processors so that they ran at an optimum usage for a given job and it would be a reasonable price.

We will return to this issue in our experiment with FTK, in section 4.3.4.

4.3.1.6 Hadoop processing local data stored across the cluster

In 4.3.1.3 we saw how MapReduce and Hadoop enables vast quantities of data to be processed. They key to this is to initiate processing at the data storage location. It would be desirable if this could be applied to a middleware intended to process forensic data. The need to transfer data before processing was the main failing in Roussev's DELV system (Roussev & Richard III 2004) as was examined in 4.3.2 and FTK distributed processing in 4.3.4.

4.3.1.7 HTCondor's Classified Advertising and Job Submission Definition

In Figure 25 – An example of an HTCondor Classified Ad file, on page 88, we saw how HTCondor allows each host on the network to define and publicise its capabilities. This allows fine tailoring of load balancing. Hadoop seems to lack this ability to allocate specific jobs to hardware with certain capabilities. Hadoop seems to be based on the assumption that it will run on standardised hardware where the capabilities of the hardware do not vary greatly. HTCondor, on the other hand, allows fine-grained allocation of tasks to hardware that fulfils criteria set by the user. From the hardware owners point of view the Classified Advertising facility within HTCondor gives fine-grained control over the availability of resources.

On the user side of this arrangement, as we saw in Figure 26 - HTCondor Job Definition, on page 88, HTCondor uses an extensive script language to initiate jobs on the cluster. This links in with the Classified Advertising facility to provide the user with a high degree of control over their job submission.

4.3.1.8 Distributed File Systems, File Striping, Access Speed and local processing

Distributed file system were first created alongside the first developments in supercomputers in the 1960s.

The major relevant developments started in the 1980s when Digital Equipment Corporation developed a file system called "Network File System" in 1984 (Sandberg et al. 1985). Another notable file systems at this period included the "Server Message Block" (SMB), or more recently called "Common Internet File System" (CIFS) used by Microsoft (2014a).

Designs for Distributed file systems have a number of goals that are similar to those of Distributed systems in general. These are evolved from the ideas identified in section 4.3.1 from the work of Enslow and Coulouris but focus on transparency.

- *Access transparency* is that clients are unaware that files are distributed and can access them in the same way as local files are accessed;
- *Location transparency*; a consistent name space exists encompassing local as well as remote files. The name of a file does not give its location;
- *Concurrency transparency*; all clients have the same view of the state of the file system. This means that if one process is modifying a file, any other processes on the same system or remote systems that are accessing the files will see the modifications in a coherent manner;
- *Failure transparency*; the client and client programs should operate correctly after a server failure;

- *Platform Transparency; heterogeneity. F*ile service should be provided across different hardware and operating system platforms;
- *Size Transparency: Scalability. T*he file system should work well in small environments (1 machine, a dozen machines) and scale gracefully to huge ones (hundreds through tens of thousands of systems);
- *Replication transparency*; to support scalability, we may wish to replicate files across multiple servers. Clients should be unaware of this;
- *Migration transparency*; files should be able to move around without the client's knowledge.

There are a large number of file systems and each had its own design objects. Successfully fulfilling one design objective can sometimes mean the file-system is very inefficient in another application setting. Identifying a specific need is the key to the correct selection of an appropriate file system. Typical criterion might be the number of files likely to be stored within the file system, the maximum or typical file size, the file naming convention or speed of access.

Choosing to use a distributed architecture implies a desire to utilise the processing power and storage facilities of the whole infrastructure. Branches of a directory hierarchy can be connected to mount points on a file system structure. This feature can be stackable so that huge structures can be built without the user, be it a program or human, having any knowledge of the actual location of the data being accessed. Andrew File System (Carnegie Mellon University 2015), GlusterFS (Gluster.org 2015), Lustre (Lustre 2015) and Parallel Virtual File System (PVFS) (PVFS 2015) all employ this architecture.

This obviously improves scale but usually implies loss of performance. Each layer of the hierarchy introduces a corresponding delay in transmission times as data is called and transferred across the network to a host for processing.

If speed is the priority, a form of data stripping, similar to that found in RAID arrays, can be used but on a much larger scale. In this case, data is dispersed across many hosts in the same way as it is across many disks in a local RAID array. This means data transfer from source hardware, probably hard disks, is concurrent and so much faster. A four disk RAID 5 array can read data at about four times that of a single disk. The IBM General Parallel File System (Schmuck & Haskin 2002) uses this this approach. Stripes are read from the storage media and combined in local memory before being available to the application program. This does mean that the data stored on individual storage media is incomplete can is of no use without the rest of the dataset that makes up the stripe.

All these file systems have developed over time and have additional features to enhance performance but recently, Ceph File system (Weil 2007) (Weil 2007), has combined both techniques from the start of the design allowing large structures of files to be built, whose data is stripped out across many hosts.

All of these architectures assume data transfer will occur when data is required for processing. In section 4.1.2, we explained that data transfer was really the core of the current problem.

In recent years, the issue of processing very large files has led to the design of file-systems like Google File System and Hadoop Apache where the data that comprises very large files is broken in chunks or blocks and stored across many storage servers while maintaining “location Transparency”. This is not stripped, as in the case of RAID or Ceph. Whole chunks of files are stored in a readable form.

4.3.1.9 Programming languages

If a program is to utilise the MapReduce model, Hadoop requires programs to be written with the MapReduce architecture in mind and to this end, it is highly recommended that it be written in JAVA. JAVA was chosen, presumably, because of its portability rather than other features of the language. We acknowledge that there is a considerable volume of software that is available for the forensic investigator to utilise that is not written in JAVA. By attempting to design a system that imposes the least restriction and expectation of the abilities of the programmer, we can maintain the openness of the system.

This has a close association with the HTCondor Classified Advertising facility. Programs written in specific languages may only be capable of running on specific configurations of hosts. This might require data to be moved, which goes against the Hadoop principle of processing data locally.

4.3.2 2004 - DELV - Breaking the performance Wall

“Breaking the performance Wall” (Roussev & Richard III 2004) is considered the base work in this area of forensic processing and is certainly the most cited. Vassil Roussev and Golden Richard were amongst the first to produce a paper that identified the increase in digital evidence as a major problem for digital forensics in the coming years. At the time, they wrote about drives of 200GB costing $165 and issued a warning that “*investigators should be prepared to handle targets with massive amounts of data*”. They identify that, because of the connection of a high power CPU to high capacity storage through a relatively low capacity IO, forensic data is a processing intensive activity. They asked the fundamental question “*In summary, the case for going distributed in forensic analysis is not fundamentally different than in any other application domain where high performance is needed. The obvious question is: should we try to use generic distributed frameworks (GDF) and adopt them for our purposes, or are we better off developing a more specialized solution?*” Based on their assessment that “*A targeted solution can be better optimized for its specific purpose* *and, hence, achieve better performance with less overhead. Generic frameworks tend to be heavyweight since they try to be everything to everyone. Usually they provide simple programming abstractions, such as distributed shared memory, that are convenient but may become serious performance bottlenecks.*" they dismiss adopting a generic distributed framework and choose to design and build a custom solution, latterly known as DELV, based around a bespoke message passing protocol and a central file store facility.

There were a number of distributed middleware systems in common use in 2004 when Roussev and Richard designed DELV. No doubt, they influenced the characteristics of their design. This design retained the idea of a central file store from which data was transferred out to be stored in the RAM of processing workers. They pointed out that if each of the hosts in a 64 node Beowulf cluster had 2GB of RAM this would provide about 100GB of RAM that could act as storage with very high IO performance.

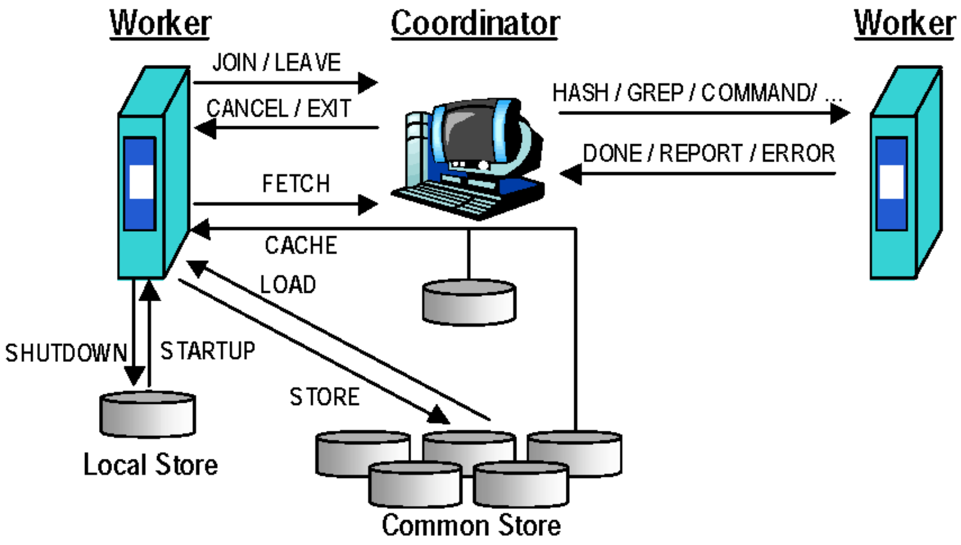


Figure 29 - DELV System Schematic (Roussev & Richard III 2004)

In a further effort to optimise the system away from the heavyweight generic solutions they dismiss standard message protocols like GENA and SOAP, saying that "*we do not foresee the communication leaving the private LAN of a forensic lab*" and instead design and implement a lightweight protocol of their own. Certain characteristics of their solution can be deduced from the commands they describe.

JOIN and LEAVE commands are issued by the worker hosts to notify their availability for accepting tasks. EXIT, SHUTDOWN and STARTUP are issued by the co-ordinator to control the worker. CACHE and FETCH are issued by the co-ordinator to instruct the worker to allocate RAM memory, retrieve a file, or files, from the central store, and keep them in the RAM of the worker. There are a series of progress commands like DONE, ERROR, REPORT, PROGRESS and CANCEL that can be used to monitor the activities of the workers. There are then a series of processing commands like HASH, GREP, THUMB, STEGO, CRACK and EXEC that actually trigger the work. These commands indicate that the system is a ‘Command and Control’ design where the co-ordinator is the central control issuing commands.

They implement their system on the ‘Gumbo-72’ cluster at the University of New Orleans. Figure 29 and Figure 30 show the system schematic and architecture, respectively

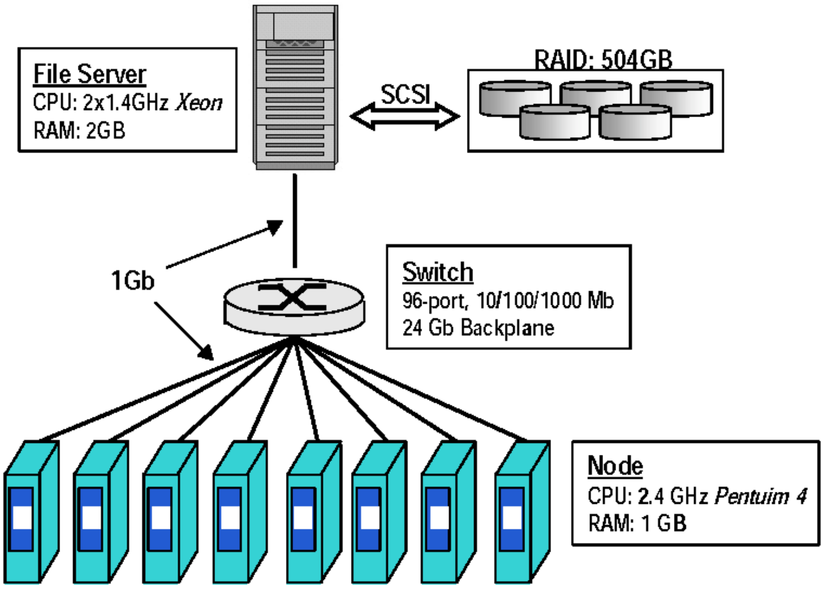


Figure 30 - DELV System Architecture (Roussev & Richard III 2004)

Their testing was based on a 6GB image that they stored on the RAID server so each of the workers was able to cache about 700MB of data. They, quite rightly, acknowledged that it is not possible to make a fair comparison with FTK at the time because

> *“None of the measurements for our prototype are directly comparable to these numbers, because FTK is performing a lot of initial pre-processing of the (forensic) image and we only have a general idea of the implementation”*

but the results were quite dramatic and certainly proved, in principle, that there were benefits from this approach.

One of the most striking observations in the previous figure is the sizes of RAM, central storage at only 540GB and their choice of an image of 6GB. Even in 2004, this may have been considered small but now, 10 years later, they seem diminutive. If we scaled this up to the technology available in 2013, we might expect 64GB RAM, a central storage of tens of terabytes and images of terabytes as well. However, one thing that would not change is the network speed. The Gumbo-72 cluster was built with one Gigabit Ethernet connectivity. Ten years later this probably would not change but if it did it would be only by a factor of 10 to 10 Gigabit networking in comparison with 32 times the RAM, 50 times the central storage and hundreds of times of the size of the forensic images to be examined.

They quite rightly identify scalability as one of the key issues in clusters but could not have reasonably predicted increases of the scale that have happened. With the data increases of the last 10 years, transferring data across the 1-gigabit network would probably negate the gains of distributed processing in DELV. Foreigners

They note that there are a number of other key issues in distributed processing in addition to simple processing power, these repeat the observations by Enslow. The most important being 'Robustness', they omit 'load balancing' and 'resource management' completely and have no mechanism by which the special attributes of workers can be matched to tasks e.g. password cracking to hosts with GPUs. This is of course understandable as GPUs where in their infancy at the time.

4.3.3 2008 – Is a Grid suitable for digital forensics?

In response to Roussev and Richard's work, Pringle and Sutherland (Pringle & Sutherland 2008) attempted to recreate the DELV system using a Globus 4 Grid system. Technology had moved on by then and the paper "Is a Computational Grid a Suitable Platform for High Performance Forensics?" states that 500GB drives were available from British retailers for £99. They tried to build a system that closely resembled the New Orleans equipment, which was surprisingly difficult although only 4 years had elapsed. With the benefit of 3 years progress, they state that they found that the Globus Grid was available as Instant Grid (Instant-Grid 2013) based on the Knoppix 5.1.1 Live Linix CD. It is often observed that system installation has become easier over the years as anyone with more than 10-15 years of Linux experience has observed. This is also true of Globus, the installation of which has become much easier over the years.

The most important observation we made was that the distribution of data from a central file store out to the workers was the main bottleneck on the system. Once the data was stored on the workers, the speed increase was dramatic.

In the Globus implementation of the DELV experiment they used about 4GB of data with about 5000 files; roughly comparable to Roussev and Richard's experiment. They tried sequential copies from a central store using NFS, and parallel copies using gsi-ftp and then grep searches from the local hard disk of each worker.

Host	files	Total Size MB	NFS Copy Time secs	Gsiftp Copy Time secs	HD Search Time secs	RAM Search time secs
H001	859	660	39	94	16	2.6
H002	978	648	36	114	17	3.1
H003	826	669	37	88	17	3.3
H004	758	654	33	90	19	3.1
H005	822	670	36	88	17	2.3
H006	838	664	37	97	19	3.4
Total	5081	3965	218	571	105	17.8

Table 10 - Data Processing Times in the Prototype Grid

From the results, in Table 10, we can see that copying the data from the central storage took about 40 seconds for each worker using NFS. This was ‘running in parallel’ and is cumulatively limited as there is only one-gigabit Ethernet connection from the storage server. When they used the Grid gsiftp this time multiplied typically by three. This is in line with Roussev and Richard’s observations about protocols based on HTTP. Once on the worker’s hard disk, a grep search took about 17 seconds, understandably, as it needs to read the data from the hard disk. However, once the first search is complete the data remains in RAM and is then searched at far greater speed of less than 4 seconds.

Their estimates at the time, where that the data transfer achieved 18MB/s over the Ethernet, the first search from the hard disk achieved 38 MB/s and the search from RAM was 222 MB/s. As the data transfer rate from even PC3-3200 RAM is about 3GB/s we believe the task became processor limited at that stage, 222MB/s being the limit of the processor to perform the grep search.

This broadly agreed with Roussev and Richard’s figures, however by using the Globus Grid they were using a full ‘professional’ system with the resources of the Globus foundation for its development. This provided installation, redundancy, monitoring and many more facilities.

4.3.4 AccessData FTK and large datasets

AccessData’s Forensic Toolkit was first released in the late 1990s. Distributed processing was introduced in version 3.0 with four workers in August 2009. It is currently the only example of a commercially available application of distributed processing in digital forensics. The following information is taken from “Divide & Conquer: overcoming Computer Forensic Backlog through Distributed Processing and Division of Labor” (AccessData Corporation 2009)

AccessData has retained the centralised storage of forensic image files, shown in Figure 31. The disadvantage of a centralised store for forensic images, as was the case in both DELV and GRID, is that the data needs to be moved across the network to be processed. It is reasonable to assume that AccessData is constrained by the need for commercial acceptance of their system and that they achieve this by retaining an architecture based up on a forensic image approach.

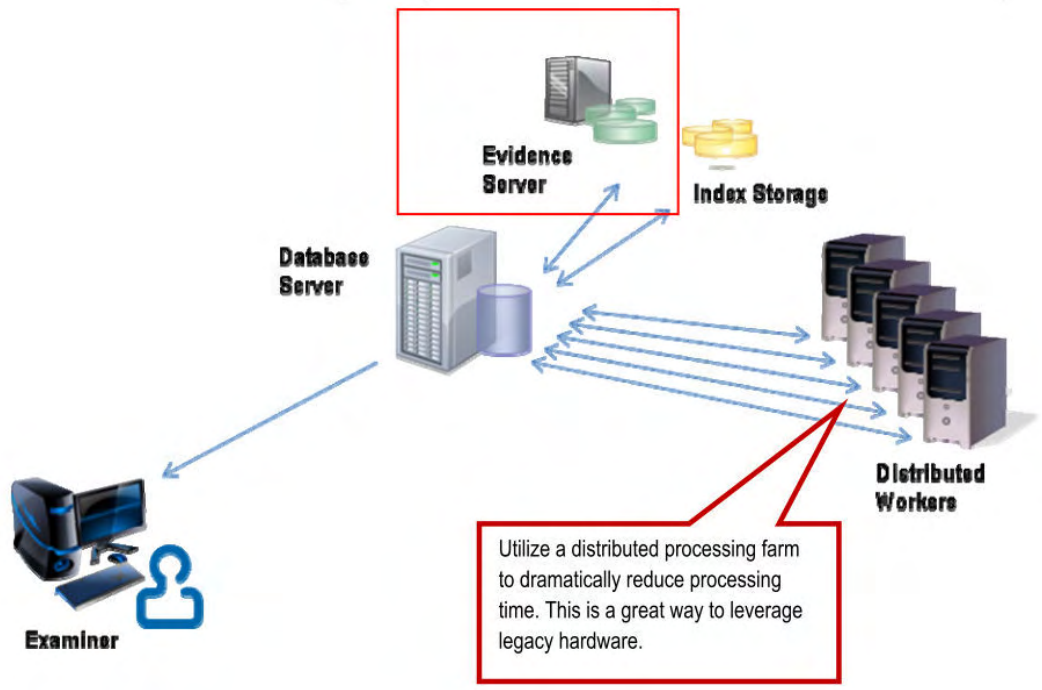

Evidence Server
Index Storage
Database Server
Distributed Workers
Examiner
Utilize a distributed processing farm to dramatically reduce processing time. This is a great way to leverage legacy hardware.

Figure 31 - The Evidence Server (AccessData Corporation 2009)

We set up an FTK system to assess the impact of the data transfer from a central storage facility, out to the data processing hubs. As we were interested in data quantities rather than speed it was only necessary to process a very small image to get ratios of data quantities. As a result, we processed an image of 358MB (375,445,069 bytes) with a mixed bag of files including photographs, word documents and excel spreadsheets.

AccessData's larger version FTK-Lab seems to exhibit the same architecture seen in their system illustration, Figure 32

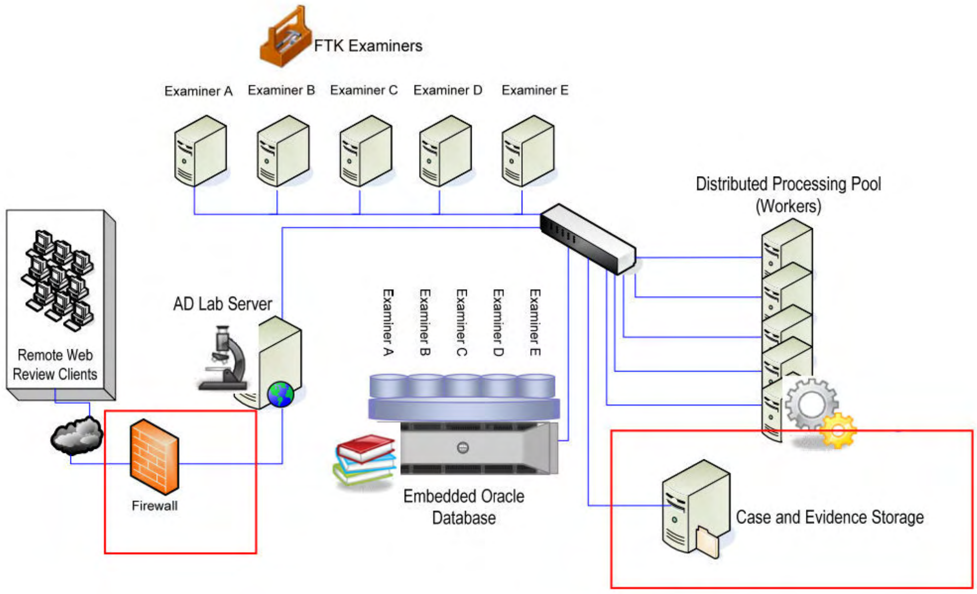


Figure 32 - Case and Evidence Storage (AccessData Corporation 2009)

To assess the impact of this central server approach we conducted a small experiment by building a four-processor installation of the Access FTK Forensic system.

Our setup consisted of seven hosts.

- The main host, GUI/DP1, is the 'user console' and hosts the primary distributed engine;
- The SQL host supports only the SQL database;
- DP2, DP3 and DP4 hosts support only the additional three distributed processing engines;
- The Case File host holds any additional files created during the analysis, for example thumbnail images and text indexes for searching;
- The Evidence host holds only the source image.

Table 11, on page 104, shows the quantity of data transferred between hosts when processing a small 350MB image that was considered typical of a memory stick.

We found that during the processing of a 358MB image, 335 Mbytes (93%) is sent to the primary Data Processing Host that also run the Graphical User Interface (GUI/DP1). Surprisingly, it sends only 29 Mbytes (9%), 18 Mbytes (5%) and 38 Mbytes (11%) to DP2, DP3 and DP4 respectively.

- Processing 335MB of data resulted in moving 480MB of data between the hosts on the network;
- While 358MB of data was processed, only 3.286 Mbytes (0.9%) of data was written to the database. Inspection of the PostgreSQL database shows that only analysis metadata is stored. The evidential raw data is held in the image;

- While 335MB of data was processed, only 39.999 Mbytes (1.1%) of data was written to the case folder. The case folder appears to hold data such as the text index files. This resulted in a case folder of 50MB – 5%;
- The analysis of 335MB of data resulted in a database of about 135MB – 40%. From Table 11 we see that only 3MB of data was sent to the database. We suspect that the difference is because of fixed length fields in the database being padded out with spaces.

This experiment suggests that FTK does not utilise the DPs anywhere near as much as it could. Currently it is about, 80%, 7%, 4%, 9%. If the distribution of tasks to the DPs was more even say, perhaps, 35%, 20%, 20%, 20% even with extra management it would improve processing time by perhaps 3 fold.

This displays the characteristics of the observation made by Dean and Ghemawat when they proposed that MapReduce be focused on Data Locality. The 'cost' of moving data from a central data server out to processing nodes often exceeds the advantage of distributed processing.

All Kbytes	Host B							
Host A	GUI/ DP1	SQL	DP2	DP3	DP4	Case File	Evidence Image	Sum of Bytes out of host A
GUI/DP1		1,196	1,464	1,693	1,164	30,983	3,101	39,602
SQL	1,284		650	522	634		2	3,092
DP2	1,324	685				2,953	258	5,219
DP3	1,342	742				2,009	222	4,315
DP4	995	661				4,053	312	6,021
Case File	800		291	244	142		2	1,479
Evidence Image	335,155	1	29,230	18,072	38,379	1		420,838
Sum of Bytes in to host B	340,899	3,286	31,635	20,532	40,319	39,999	3,896	480,567

Data transferred			Example
from a specific Hosts **A**, listed on the left, to a specific Host **B**, listed across the top	**A to B**	is denoted as volume in Kb.	**29,230** kb bytes were transferred from "Evidence Host" to "DP2"
Total transferred into a specific Host **B** regardless of the source	**All to B**	is denoted as volume in Kb.	**39,999** kb were received by the Case File regardless of the source
from a specific Host **B**, listed across the top, to a specific Host **A**, listed on the left	**B to A**	is denoted as volume in Kb.	**30,983** kb bytes were transferred from "Case File" to "GUI/DP1"
Total transferred out of a specific Host **A** regardless of destination	**A to All**	is denoted as volume in Kb.	**39,602** kb were sent by the GIU/DP1 regardless of destination
All data transferred	**All to All**	is denoted as volume in Kb.	**480,567** kb passed across the network

Table 11 - FTK Distributed Processing Data Transfer Values

4.3.5 Running Programs

Handling program execution is a complex problem. There are a number of characteristics to consider that include whether:

- A program may, or may not, require user interaction either before initiation or possibly during processing. Parameters may need to be selected before the processing is started;
- A program may result in the creation of one or more new data files such as the creation of thumbnail images from high-resolution image files;
- A program may result in the amendment of one or more existing data files such as the case when a database is amended because of processing.

Ideally, programs that are run on FCluster would be initiated from a command line and either have parameters on the command line or have parameters retrieved from a file whose location and name are hardcoded into the program or are supplied by the command line. This is common practice within the Linux community. This situation is easily accommodated by the adoption of a job submission control file such as was reported to be implemented in HTCondor. HTCondor's Job Definition file has parameters such as "input=", "output=" and "error=" to define data inputs and outputs to be used during execution. FCluster would use the same technique. It is beyond the scope of this prototype design to define exactly how this would work but they could connect to files that are then delivered to the user, they could be stored within a database or many other options.

The widespread adoption of Microsoft Windows has encouraged a style of user interface where the user is presented with a program container, often Multiple Document Interface (MDI), and they then initiate a task with a selection from a menu, a key press or click on an object on the screen. The user is sometimes then presented with a dialog box into which they enter, or select, options and the task is initiated with a click on an appropriately labelled button. The code for the user interface and the code for execution are contained within the same module and are, effectively, inseparable.

4.3.6 The wider forensic community will continue to produce useful software

Although there may be greater weight placed on evidence revealed when established software from known sources such as Guidance Software and AccessData, we should acknowledge that the wider research community has and will continue to develop productive software.

Any similar development should allow this to continue in a way least taxing for the programmer but still allow them access to the leverage of power in new forensic systems.

4.3.7 Open Architecture Information Systems OAIS - ISO 14721:2012

FCluster, the subject of this thesis, was designed and written in the summer of 2013. Six months later, it was presented as a paper to the Digital Forensic Research Workshop Europe 2014, with submission in December 2013. During the review process, two reviewers pointed out an existing system of information archiving used to store data from space exploration over the last 2 decades. This information was not known at the time of designing FCluster but is highly relevant in evaluating FCluster in chapter 7.

The Open Archival Information System (ISO 2012a) was originally designed to accommodate the huge amount of diverse data originating from observation of the terrestrial and space environments.

OAIS proposes a learning model, shown in Figure 33 - Data to Information in OAIS, to represent the transformation of data into information.

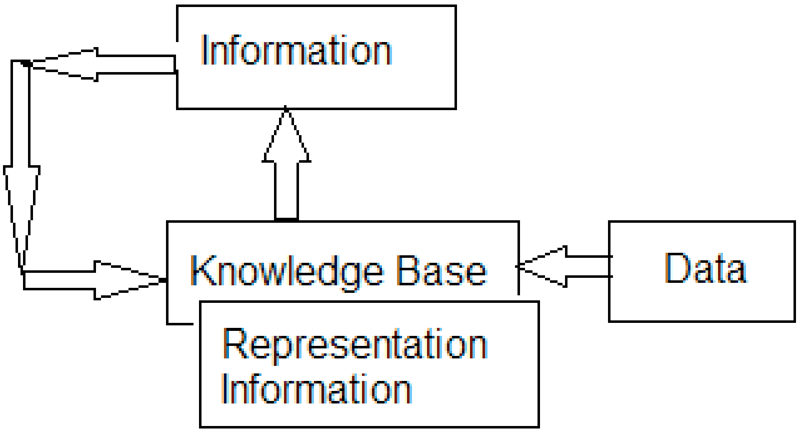


Figure 33 - Data to Information in OAIS

An OAIS archive has three distinct entities Producers, Consumers and Management - Figure 34 - OAIS Environment Model.

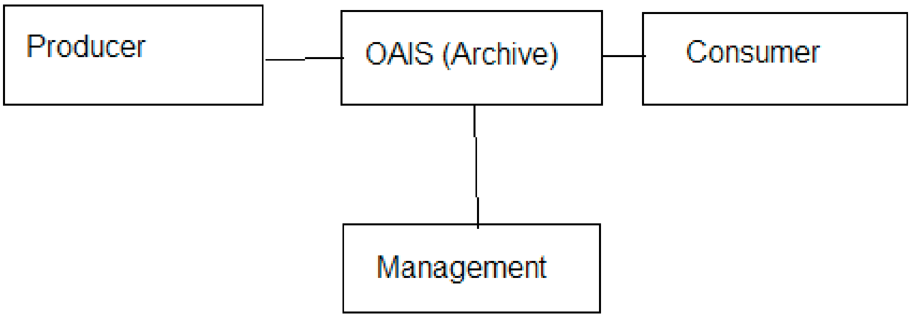


Figure 34 - OAIS Environment Model (ISO 2012a)

Data is generated or gather by a "Producer". In our domain, this could be the suspect, victim or DEFR and is packaged into Submission Information Packages (SIPs). The Producer would initiate its "Ingestion" into the system where it is processed and stored as Archival

Information Packages (AIPs). A consumer, the DES, can issue Queries against the archive which may illicit a response resulting in a list of AIPs. The Consumer can then order one or more AIPs resulting in Dissemination Information Packages (DIPs) being formed and sent to the Consumer. All this is subjected to the control of The Management whose job it is to develop and implement policies that govern data access and quality. A system schematic is shown in Figure 35.

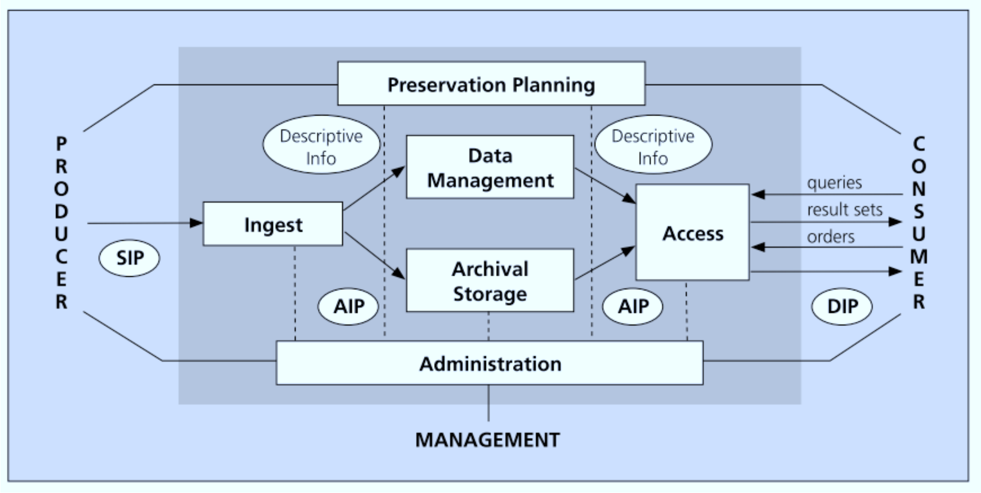


Figure 35 - OAIS Functional Entities (ISO 2012a)

OAIS describes a transactional system where data is deposited by the producer and the consumer issues orders, to which the system responds by issuing dissemination packages to satisfy that request, Figure 36.

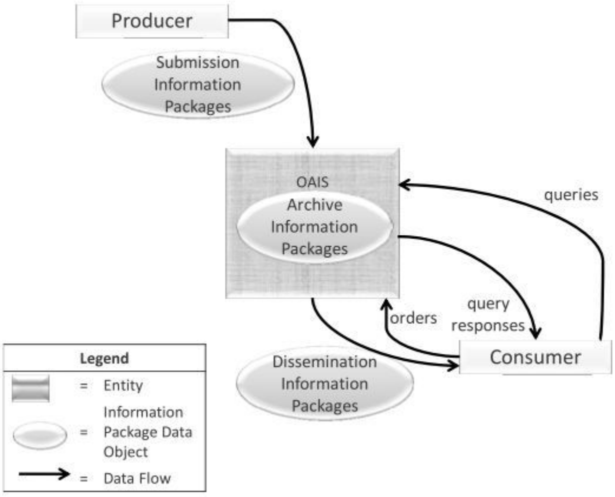

Producer
Submission Information Packages
OAIS Archive Information Packages
queries
query responses
orders
Consumer
Dissemination Information Packages
Legend
= Entity
= Information Package Data Object
= Data Flow

Figure 36 - OAIS Archive External Data (ISO 2012a)

At the stage of acuisition, when the original data is packaged into SIPs, additional Content information is gathered and attached to the core data to provide context and provenance.

The creation of a co-responding AIP implies the creation of Preservation Description Information (PDI). The PDI comprises of data that defines reference information about its source. For example, this could be a crime case number. In the case of a Digital Evidence Container, the data's Context Information would be information about which media it came from. The AIP contains Provenance Information, which would be our chain of evidence information. Finally, it contains Fixity Information that corresponds to the integrity data represented by MD5, SHA-1 cryptographic hashes.

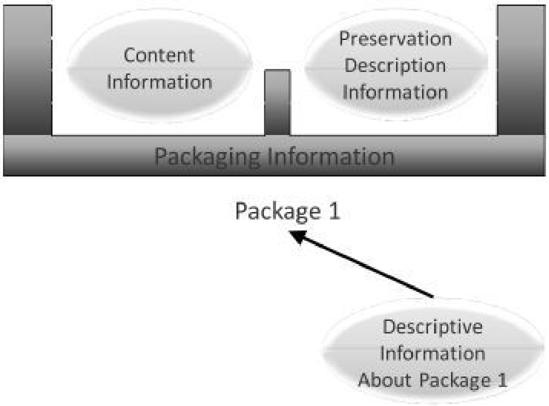

Content Information
Preservation Description Information
Packaging Information
Package 1
Descriptive Information About Package 1

Figure 37 - Information Package Concepts and Relationships (ISO 2012a)

In OAIS information packages consist of two sub-components, shown in Figure 37 - Information Package Concepts and Relationships, one contains meta-data describing the nature of the raw data and the other, the actual data. Associated with this is descriptive meta-data that relates to the information gained from the data because of the data to information cycle described in Figure 33 - Data to Information in OAIS

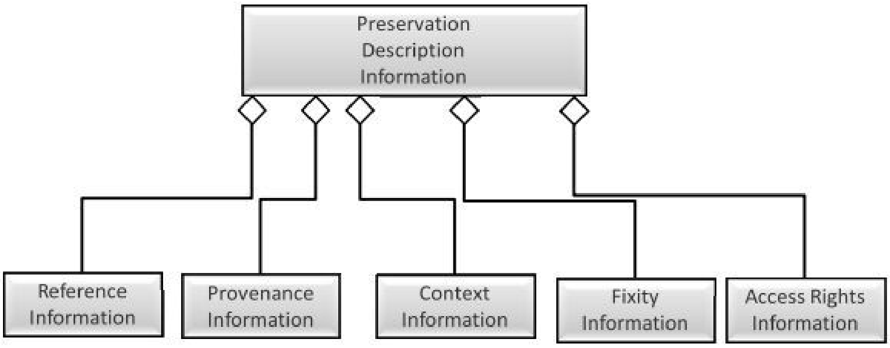


Figure 38 - Preservation Description Information (ISO 2012a)

OAIS has its own data format - Metadata Encoding and Transmission Standard (METS). METS provides a means of associating all the metadata of an object, including the object's relationship with other objects, within the object. This is shown in schematic form in Figure 38.

A METS definition is in the form of an XML Schema that is extensible to whatever is needed. The METS Editorial Board recommends a number of formats but these are not the limit. The US Library of Congress (The Library of Congress 2014) is developing a series of METS which is offers as a base for standardisation and is the official web site. These can be amended or extended as required.

It seems that there are two key differences when a comparison is made between OAIS and existing forensic systems. Firstly, in OAIS it is assumed that there is a format change between the SIP used in the submission stage and the AIP used in the storage stage. Currently, the accepted best practice in digital forensics is that, where possible, an image is taken and stored within the forensic system and although it may be analysed and the results stored in a database, reference is always made to the original image. Secondly, there does not seem to be a facility WITHIN OAIS for processing. Data always needs to be Ordered and Disseminated to a location where processing takes place, with the resulting information being sent back into OAIS.

4.4 Interim Discussion

It should now be clear that current processing models, primarily single PCs or PCs on a network, will struggle to cope with the increase in data volumes being experienced in digital forensics. This is a problem of imbalance between system components and is not unique to digital forensics. In the last decade, Hadoop has enabled analysts to tackle vast quantities of data; the techniques of data-mining are already revealing quite amazing patterns in huge datasets. This is a new domain of research – data analytics.

A similar new solution is needed in digital forensics. There are several on offer but only one that uplifts all the others with wide benefits.

Data reduction presumably requires mass data to be pre-processed in order to reduce the volume; this increases processing time. Data mining is usually statistically based, which is well known for being processor intensive; again increasing processing time. Larger single processors are disproportionately expensive. Currently improving case management usually implies using a triage approach that could be viewed as selectively ignoring or discarding evidence or entire cases.

Distributed processing is a platform not an actual application solution. It can provide us with greater storage, shared workspaces, more raw power to enable data reduction and provide a platform for more complex mathematical text processing algorithms.

The history of multiprocessors clusters and distributed processing goes back to the 1960s. There are many examples of cluster and distributed systems being successfully deployed to solve problems. These systems are at their best when they have a clear design target and the features are focussed to that end. Nimrod is design for parameter sweeping. Weka is design for data mining. HTCondor maximises processor utilisation. Hadoop is one of the latest to join the collection. Hadoop could be used for forensics but it is really designed for analysing log files. Anything else built on top, hBase for example, dulls its cutting edge.

If we were to set our sights higher than adopting a general purpose solution and consider what might be required of a system specifically designed for forensic processing there are lessons to be learned from Hadoop. Key to Hadoop success is its file system HDFS. If it was possible to design a file system with the specific needs of handling and processing data for digital forensics then it could be the base of an acceptable new architecture. Designing a file system from first principles can be done but is complicated. Recently FUSE file systems have provided the opportunity to develop, effectively, custom file systems based upon the sound foundations of well-established systems like ext3/4.

Since the file system covers all data access, it is possible to create a far-reaching control structure, a middleware that has complete authority over data storage, access and integrity. FUSE file systems have been proposed within digital forensics previously. There are several FUSE file systems that have features of interest and so a fusion and extension of these could provide the foundation we need.

We believe that because of the increasing size of media, the adoption of cloud systems and storage, the notion of the 'image' as we understand it today, will become increasingly untenable. Partial and selective imaging will become more common and ultimately the norm. Evidential data will need to be stored as separate files across the system. Consequently, we must expect tens, if not hundreds of millions of files in a system that comprises the information previously stored the contents of many forensic images. We believe file formats, like AFF (4.6.2.8), will predominate in the next few years.

The practice of taking a forensic image as one big file does not align with the dispersed concurrent nature of distributed processing. To be successful we need to devise a new paradigm of imaging. We have a lead from recent work by Roussev on LOTA, reviewed in section 4.6.1.3, but this can be improved.

Chain of evidence procedures are well established in the real, analogue, world. Documentary proof is accepted, anti-tamper packaging is used and evidence is held in a secure locker. This is an end-to-end solution. We have treated the computer system as we would a single room secure storage locker. Until now, the computer system has been seen as a single place we store and process data. This is certainly true for a PC sitting on a desk but we seem to have slipped into the same thinking when we connected these via a local area network isolated from the Internet it's the same situation. Even if this was true, surely, this thinking cannot continue into a distributed processing architecture.

In the next few sections, we cover the additional areas of FUSE file systems, Imaging and data processing prioritisation needed to build a true concurrent distributed system.

4.5 Existing FUSE File Systems

4.5.1 Introduction

Although the MapReduce paradigm is the key to Hadoop's success in processing, the Hadoop Distributed File System is the key to its successful implementation.

There are many distributed file systems available. As we saw in section 4.3.1.3, each is designed to fulfil a specific set of requirements identified by the designers. No single solution is successful in every circumstance; there are always compromises

Designing a file system from scratch is a difficult task that has, relatively recently, been made rather easier. With its origins in the mid-1990s, the **F**ile **S**ystem in **U**s**E**r space (FUSE) project started in 2004 (FUSE 2014). It was officially merged into the Linux kernel in version 2.6.14 in 2005 (Linux Kernel Newbies 2007). Figure 39, taken from the FUSE web site, shows the schematic layout of the FUSE file system.

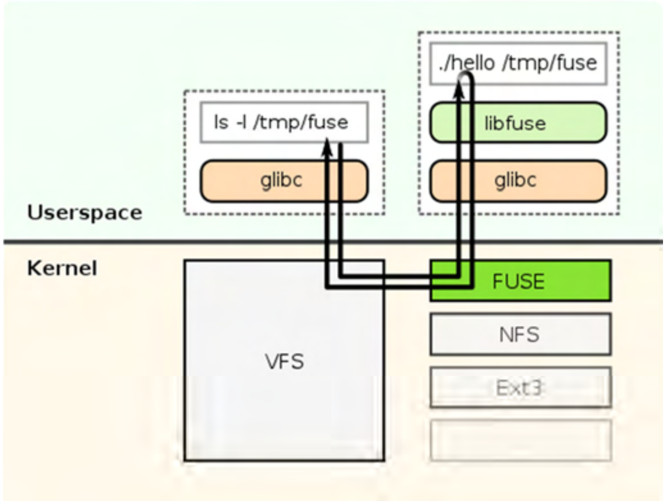


Figure 39 - FUSE File System Schematic

FUSE file systems are built on top of existing, well established and tested, file systems. A file-system request from an application program, in the figure this is "ls –l /tmp/fuse", is normally sent to the Virtual Files system which in calls the set of subroutines for the specific file system. If "ls –l/tmp/fuse" is used on a Linux system, it may well call routines in the ext3 library that is now built in to Linux. FUSE intercepts these calls and adds a link in the chain that is processed in user-space. In this way, additional code can be added to alter the function of the storage system.

Under normal operation (Figure 40 - FUSE File System dataflow), application programs, on the far left, can make any of a set of 42 operational calls to the Virtual File System. These calls are passed on to the appropriate code capable of accessing the file system format used on the storage media.

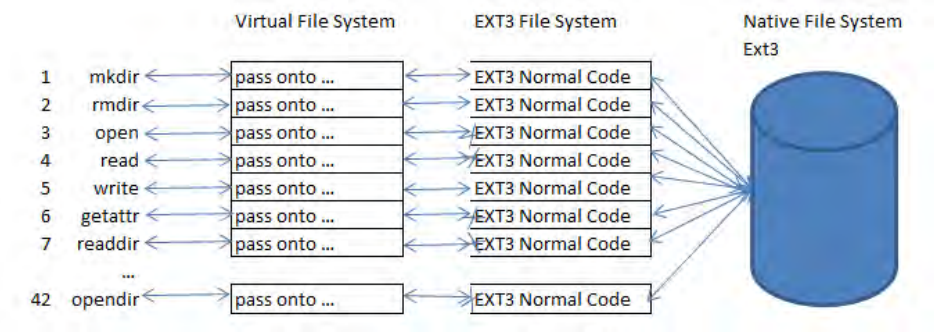


Figure 40 - FUSE File System dataflow

A File System in User Space allows an 'ordinary' user with non-system administrative user rights to load and run code which intercepts these calls as they pass between the Virtual File System and the native format code - see Figure 41 - FUSE File System with User code inserted.

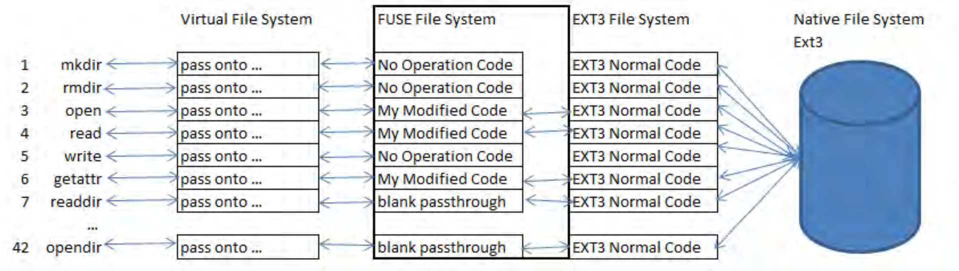


Figure 41 - FUSE File System with User code inserted

Essentially this means that existing file systems can be heavily modified in their operation and that application programs have no 'knowledge' that they are accessing a FUSE file system. The application programs do not need to be modified in any way.

In Linux FUSE, file-systems are mounted by commands similar to the following:

```
$mount /dev/sda1 /mnt/mydata
```

In which the partition /dev/sda1 is mounted at the /mnt/mydata mount point. The mount command detects the file system on the media by its format number in the partition table and employs the appropriate code. FUSE file systems work in very similar ways calling FUSE code instead.

The success of the FUSE approach is demonstrated not only by the existence of more than 50 such file systems but the huge, and imaginative, diversity of implementations. FUSE has already been used in digital forensics. There is a FUSE file system interface for the AFF forensic format and Richard et al. (2007) and Marziale (Marziale 2009) introduced the Forensic Discovery Auditing Module (FADM) as a FUSE file system.

The following FUSE file systems are of particular interest in this project.

4.5.2 MySQLfs

MySQLfs (Brancatelli 2014) is a FUSE file system that stores file system data and meta-data entirely within a MySQL database. The component parts of a native file system, like the NTFS or EXT3/4, are mimicked with tables within a database. Calls to the VFS, as described in 4.5.1, are completely intercepted by the MySQLfs code. MySQLfs provides the full capabilities of read/write/create/delete. It does not support record or file locking.

The location of the MySQL server is not limited to the local host. A remote MySQL server would only be apparent in that the connection speed presumably would have an effect on the response of file system operations.

The database in MySQLfs achieves its functionality with three tables.

- Inodes table: which stores most of the data found in the $MFT file in NTFS
- Tree table: which holds the relational data that describes the hierarchy of files and folders that make up the NTFS file system
- Data_Blocks table: This mimics the clusters on the file system that are used to store the data.

There are no tables to mimic the cluster allocation table as this data is held within the data_blocks table fields.

MySQLfs is invoked with a command line such as:

```
$fusemount   -u myname:password  \

 –ohost=10.0.0.1 \

 –odatabase=myfilesystem \

/my/mountpoint
```

This would append the contents of the ‘myfilesystem’ database on the MySQL server running on 10.0.0.1 at the file-system point /my/mountpoint. The mount will conform to any higher-level access control options.

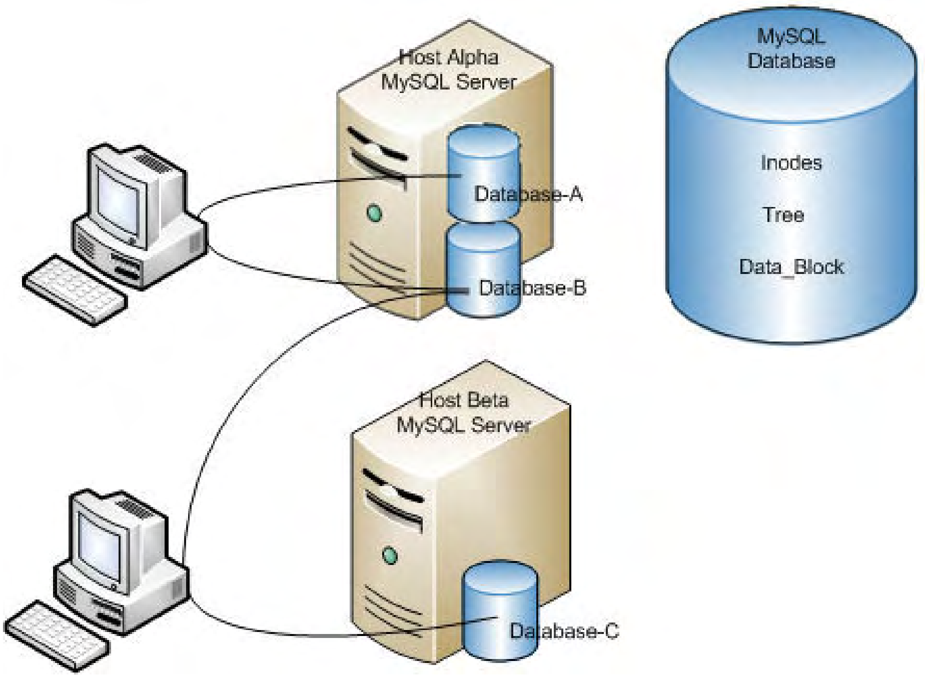


Figure 42 - mounting MySQLfs

MySQLfs can store only one file system in a single database but it can have more than one database on a MySQL server. A host may simultaneously access more than one MySQL server. Figure 42 shows 2 clients mounting the contents of three databases.

4.5.3 curlFTPfs

curlFTPfs (Robson 2013) uses the libcurl (MIT 2014) library (Common URL) to enable a connection to an number of servers types, principally ftp, to be mounted onto a host file-system as an extension to the host file system. In fact, curlftpfs support many other data transfer protocols including SSH, SMB and HTTP/HTTPS. Connections can be established across SSL and TLS encrypted links by setting options in the command line parameters. curlFTPfs substitutes common ftp commands like *get* and *put* in place of file system read and write to achieve an apparently seamless mount of a remote ftp server connection without using a conventional ftp client. CurlFTPfs is limited to a single ftp server connection per mount.

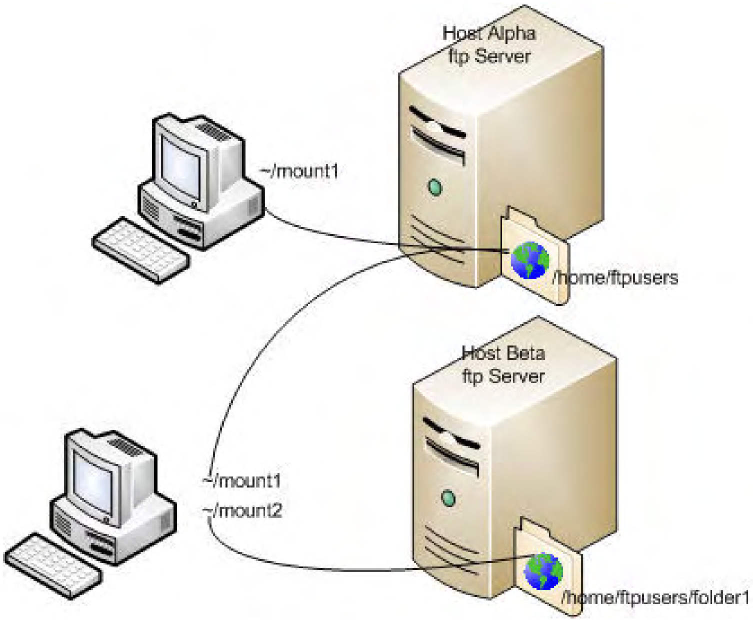


Figure 43 - Mounting curlFTPfs

CurlFTPfs has a command line such as:

```
$>curlftpfs    ftp://10.0.0.1/  \
      –o user=username:mypassword  \
   /my/mountpoint
```

This mounts the entire contents of the remote ftp server, hosted on 10.0.0.1, using the access “username” and “mypassword” to the directory /my/mountpoint. The mount will conform to any higher-level access control options and the read/write constraints of the ftp server.

A connection to only one ftp server is allowed per mount but a user can have many simultaneous mounts, each to a different servers. See Figure 43 - Mounting curlFTPfs

4.5.4 eCryptfs

eCryptfs (Hicks et al. 2013) introduces encryption for data stored on the media surface.

eCryptfs mounts a folder, which can be from a local or remote file-system, in the client's user space. When data is written into a file contained within the mount, it encrypts on the fly and writes the encrypted data to the media surface. When reading, the encrypted data is read from the media, transferred to the users RAM, often across a network connection, and then decrypted. Similarly, data due to be written is encrypted in RAM and then transferred to the storage media. In this way, all data is encrypted during transmission.

eCryptfs' command line is typically:

```
$>mount -t ecryptfs [lower directory] [ecryptfs mount point]
```

Where [*lower directory*] is the actual media directory, which could itself be a mounted file-system, and [*ecryptfs mount point*] is the mount point for the decrypted data. See Figure 44 - Mounting eCryptfs.

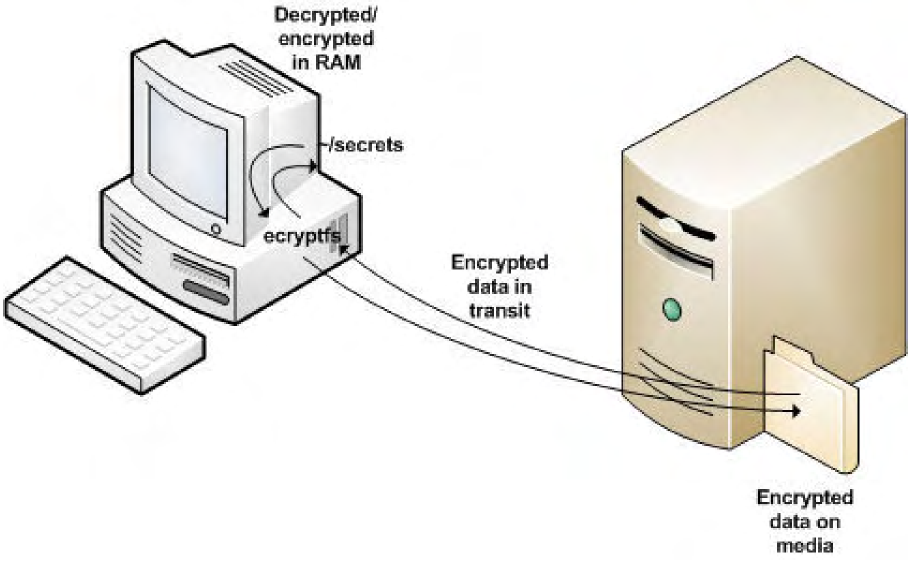


Figure 44 - Mounting eCryptfs

4.5.5 Loggedfs

Loggedfs (Flament 2013) monitors access to files in the mounted folder. Loggedfs intercepts any access to any of the files and writes out an audit log to a text file. Loggedfs give fine-grained control over logging using definitions held within and XML format text file, see Figure 45 - Loggedfs control file. Files can be included and excludes by wildcard definitions but level of logging cannot be changed and is quite detailed.

```
<?xml version="1.0" encoding="UTF-8"?>

<loggedFS logEnabled="true" printProcessName="true">
  <includes>
    <include extension=".*" uid="*" action=".*" retname=".*"/>
  </includes>
  <excludes>
    <exclude extension=".*\.bak$" uid="*" action=".*"
retname="SUCCESS"/>
    <exclude extension=".*" uid="1000" action=".*" retname="FAILURE"/>
    <exclude extension=".*" uid="*" action="getattr" retname=".*"/>
  </excludes>
</loggedFS>
```

Figure 45 - Loggedfs control file

Loggedfs' command line is typically:

```
$>/usr/bin/loggedfs –fp –c /my/loggedfs/conf.file /etc
```

Which would initiate logging of any access to the file in /etc and below according to the rules in /my/loggedfs/conf.file See Figure 46 - Mounting a Loggedfs file-system.

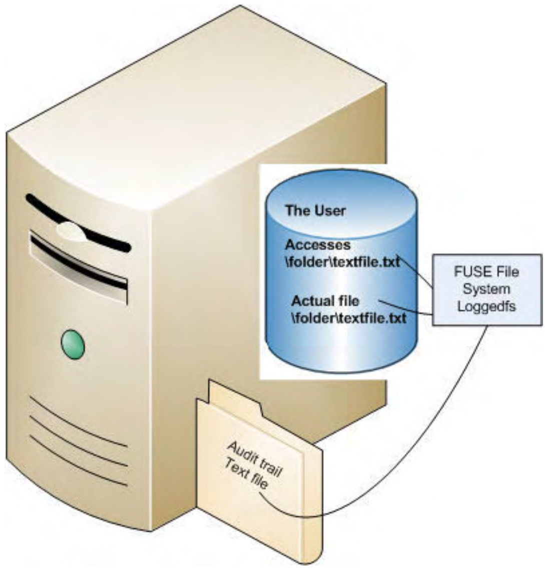


Figure 46 - Mounting a Loggedfs file-system

4.5.6 ROfs

ROfs (Keller 2014) is a FUSE file-system that intercepts any call that would result in a write action and passes through a NULL function which does not generate an error. Mounting a file system as Read Only has always been available in Linux but this makes it transparent and open to inspection.

ROFs's command line is typically:

```
$>mount -t rofs readwrite_filesystem mount_point
```

Figure 41 - FUSE File System with User code inserted" shows the use of NULL code.

4.5.7 Affuse

Affuse (Cohen et al. 2014) is a Fuse File-system that allows AFFx files to be mounted as a read-only raw files. This allows 'AFF unaware' programs to access the contents of an AFFx digital evidence container. It does not appear to have been updated for the latest AFF4 file structure. Affuse is an example of a Fuse File system that unpacks a container file to expose its contents.

4.6 Imaging

4.6.1 Existing techniques of imaging and data acquisition

4.6.1.1 Introduction

One of the conclusions of the previous research was that, to be effective, assurance needs to be implemented across the whole system. Current best practice requires that a forensically sound image is taken, transferred, stored and then processed. The success of a distributed system hangs largely on the effective implementation of concurrency. Although concurrency can be implemented at the processing stage, we are prevented from achieving this in a satisfactory manner in the acquisition stage by the linear nature of imaging. With larger and larger media, this linear process is taking longer and longer but linear imaging is not the only replication method available. We need to reconsider forensic imaging.

At a fundamental level, data can be stored as a continuous linear stream or as discrete blocks that store the data in defined sections. Transfer of data from streams to block storage is called *blocking* and the reverse is called *streaming*. Almost all storage media on general purpose computer systems with Windows or Linux operates as a block device. Even tape devices store data in blocks. Data may be a stream of bytes but it is subdivided into regular blocks. Most media found in modern devices have a block size of typically 512 bytes, although this is not the only size. Subsequently, one terabyte infers two billion blocks. Upon this, further layers are built.

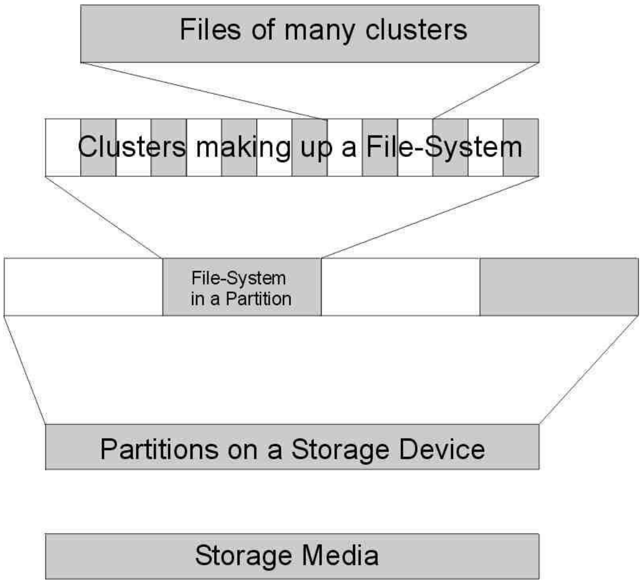

Files of many clusters
Clusters making up a File-System
File-System in a Partition
Partitions on a Storage Device
Storage Media

Figure 47 - Block Storage Abstraction

Figure 47 - Block Storage Abstraction shows the layers of abstraction built up upon the storage media. At its lowest level, the storage media is made up of electronic and magnetic components. Storage media is usually divided into partitions that usually contain file-systems. File-systems are made up of clusters that make up files, which contain data.

4.6.1.2 Linear Imaging

We will refer to the current best practice approach to forensic imaging of storage media as *linear imaging*. This technique originated in the UNIX utility dd but persists in most imaging programs currently available and trusted as being forensically sound. A bit/byte for bit/byte copy of an input device is reproduced to an output device. This takes no account of any structural data, including files-system formatting, on the media. Imaging usually starts at block 0 and proceeds one block at a time to the end. It operates directly at the storage device level.

Linear imaging has one clear advantage; speed. It does not attempt to make any sense of what is being copied and so runs as fast as is possible. This is also very useful when the data structure of the information on the media is not known. For example, many digital video recorder systems (DVRs) write directly to the media and do not have a recognisable file system. By reading directly from blocks on the media, with no interpretation of the data, we can assume that nothing will be lost in translation between the source and destination. Because no interpretation is undertaken, this technique is less prone to error, when errors occur they can normally be recovered. In addition, this provides a degree of separation in tasks between acquisition, by the DEFR, and investigation, by the DES. The DEFR does no interpretation; they collect. The DES does no collection; they interpret.

The major drawback is that it is unlikely that any data processing can be started before the linear imaging is complete. It becomes a major problem when media size is increasing. Multi-terabyte media can take many hours to complete the process. Some imaging software offers the ability to scan the data as it passes across the copy for sequences of bytes, in other words text strings but little more.

4.6.1.3 LOTA – 'Bingo'

Roussev, Quates and Martell (2013) propose a new imaging strategy "Latency-optimized target acquisition" (LOTA). We choose to name this Bingo imaging because the strategy is similar to what happens during the popular game of Bingo. From their paper:

> *"The rationale of the system is simple before we start cloning a target, we parse its file-system metadata to build an inverse map of data blocks to files. After that, we start reading the disk blocks sequentially from beginning to end, and use the map to reconstruct on the fly the files whose contents have been acquired. Once a file is complete, that is when all of its component clusters have been imaged; it is made available through the regular file-system interface to file processing tools. Disk cloning proceeds in parallel"*

The most important part of this technique is that it uses the file system to improve the imaging process and so is able to initiate processing as soon as the imaging process gets to the final cluster that holds the data for the specific file.

They acknowledge that a failing in this system is that if evidence exists in a file that, by chance, is located at the end of the disk; it will not be acquired until the very end of the process.

4.6.1.4 'Mosaic' Imaging

Another approach (Farrell 2014) recently patented, is where each block is read from the source and then stored together with its block number on a collection device. This mass of labelled blocks can then be reassembled later in the calm of the lab. This technique allows a large device to be imaged across to multiple small devices that can be combined to form a larger destination that may, or may not, be larger than the source device. Figure 48 and Figure 49 show acquisition and reassembly respectively.

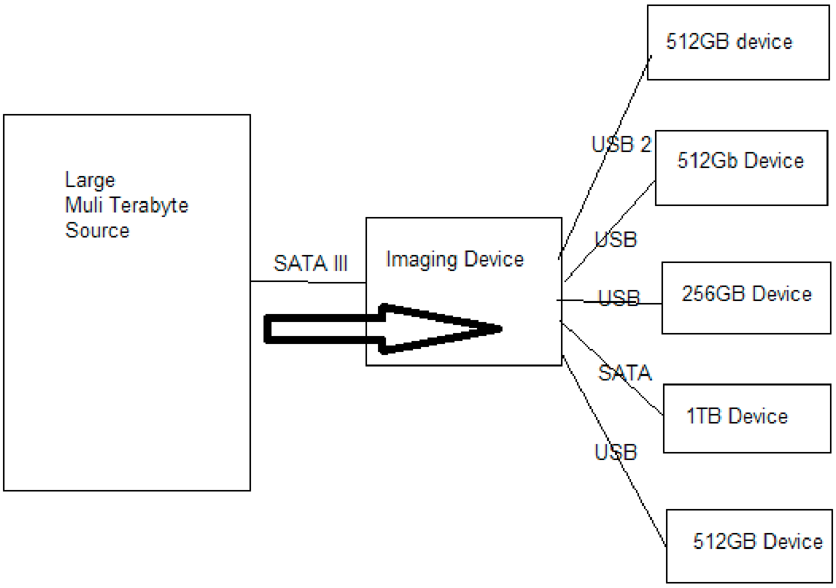


Figure 48 - Mosaic Acquisition

The main advantage is technique is that the imaging process can run through the file system following the files so 'valued' data can be acquired first. If the process is interrupted, it can still yield some useful results as it can recreate, at least, a partial image from the chunks.

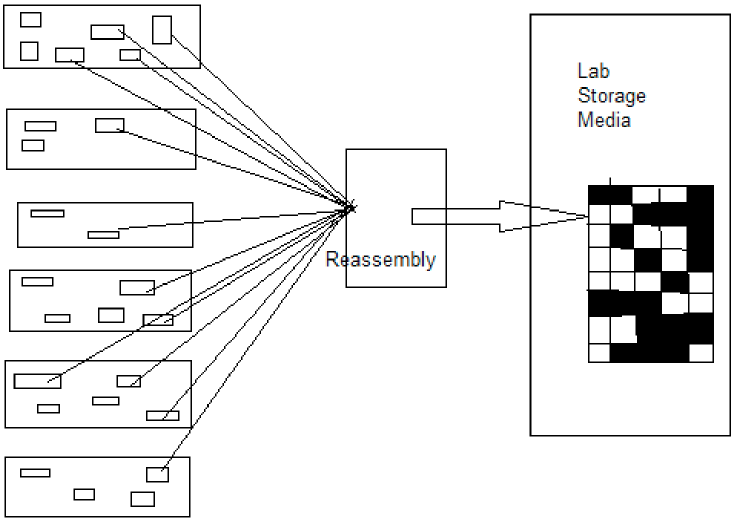
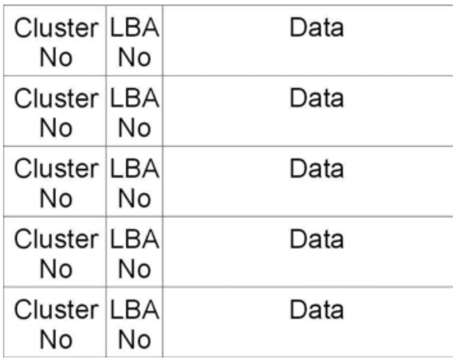


Figure 49 - Mosaic Reassembly

As this is a new patent, the exact details are not yet published in full and so it is unclear as to the exact format of the data on the capturing devices. If Mosaic created a file for each cluster, then it could mean that, as many new files are there are clusters could be created, meaning billions of small files. It is more likely that cluster number, data pairs are appended to a sequential file on each output media. The original LBA underlying the cluster could be saved as a triple, shown in Figure 50.



Cluster No	LBA No	Data
Cluster No	LBA No	Data
Cluster No	LBA No	Data
Cluster No	LBA No	Data
Cluster No	LBA No	Data

Figure 50 - Mosaic probably stores triples of data

This technique allows solid-state memory, like SD cards, which often has lower write speeds to be used in parallel for output. Having multiple destination devices could help the throughput of data by more closely matching the faster read speed from the source with the slower write speeds to the multiple destinations.

We have identified three fundamental data processing techniques for imaging. Regardless of acquisition technique, there are a number of formats in which the acquired data can be stored.

4.6.2 Current Forensic Imaging and Storage Formats

4.6.2.1 Introduction

Within digital forensics, we give great weight to the notion that all work is done on a whole copy of the image that was taken at acquisition time. We gain assurance from this because it is unaltered from acquisition time and we can prove that by reapplying the cryptographic hash process that was done at acquisition time and checking the result.

There has been extensive work on the design of file formats for forensic data and there have been many proposals for Evidence Storage Formats, though none, with the possible exception of AFF4, has been designed with the specific requirements of distributed storage and processing in mind. In section 4.3.1.3, we saw how the Hadoop File system stores data and how it, effectively, supports the application of the MapReduce processing model from which Hadoop gains so much power. However, the data chunking technique used in HDFS effectively prevents it from supporting random access across a file within the MapReduce Model. This is a significant problem with current formats used within Digital Forensics is that they are all designed with the assumption random access is efficient, or at least possible, within the file. Whole image files are now so large that this is not the case.

OAIS (ISO 2012a), see section 4.3.7 on page 106, makes a distinction between the format used for Acquisition – the Submission Information Package (SIP), the format used for storage – the Archival Information Package (AIP) and the format used for Dissemination of data – the Dissemination Information Package (DIP) and in doing so, acknowledges that one format may not suit all situations.

This raises a question about the suitability of currently defined data formats used in digital forensics when they are used in a distributed storage and processing system.

A number of forensic image formats currently exist (forensicswiki 2014c) (CDESF 2006). They fall into a series of evolutionary categories.

- A simple bit stream image of a single storage device or individual partition;
- As above with the addition of meta-data and/or compression and/or encryption
    - There are two variants in this category; searchable and non-searchable. Where the whole file is encrypted and/or compressed, it is not searchable without decrypting and/or decompressing the whole file. Later variants encrypt and/or compress in blocks of, typically, 32 kb allowing sub-sections to be processed without the overhead of decrypting and/or decompressing the whole file;
- A complex hierarchical structure that is able to contain multiple objects that can be entire devices, file-system images, individual files and extraneous data. These formats are often extensible.

Imaging is the subject of an authoritative testing program established by NIST described in (NIST 2014).

4.6.2.2 RAW image format - dd

The earliest image storage format is that of a simple binary file representing the original; this was often creating using the Unix program dd (2014).

During the default imaging acquisition process with dd, there is normally no other processing. The dd program simply reads blocks of data from the source and writes them to the destination. This has the advantage of removing interpretation from the acquisition process and so enhancing assurance by reducing the risk of an accusation of collusion and tampering with the data at this stage. All analysis and interpretation is deferred to the investigation stage when the data is processed by different staff.

The action of dd command line can be modified by supplying parameters to enable the generation of running cryptographic checksum, typically MD5 or SHA-1 but this is not stored within the resulting image. Any extra meta-data, for example a cryptographic hash needs to be kept separately. Dd writes out to one single file. This means the destination has to be bigger than the source.

In the last few years, this ‘dumb’ process has been extended to include key-word searches as the data is passing through the imaging process. This provides a simple method to alert DEFRs to significant data at the crime scene.

dcfldd (Harbour 2014) is an enhanced version of dd with enhanced functionality for digital forensics including the ability to split the destination into many regular sized blocks.

4.6.2.3 GFzip

A major problem with the dd format is that the resulting image files are the same size as the original media. These image files can be compressed using programs like zip or gzip but when this is done the ability to access data in a random fashion is lost without decompressing the whole file. In addition, using zip and gzip to compress a file, works on the whole file and so requires a second pass at the data, which increases processing time. GFZip (Meijer 2014) addresses this issue by allowing random access within the compressed file without the need to unpack the whole file. In addition, it adds multi-level SHA256 cryptographic integrity guards. GFZip development has been suspended in favour of the features of the AFF format covered in section 4.6.2.8

4.6.2.4 ProDiscover

ProDiscover (ARC Group 2014) is an open format which adds a finite set of meta-data that includes, for example, the name of the technician, a description and the date and time the image was captured, to a bit stream type image of a single file system. It can be seen as an extension of the dd format.

4.6.2.5 SMART Expert Witness Format and Guidance Software’s EnCase Image File

The SMART Expert Witness format has many similarities with Guidance Software’s Expert Witness Format version 1 (forensicswiki 2014a) as they are both originally derived from the same authors. Guidance Software’s EnCase (forensicswiki 2014b), commonly known as E01, adds integrity assurance by adding a checksum every 32KiBytes but does not provide any form of error recovery.

4.6.2.6 DEB (QinetiQ)

After about 2005, all development work on digital evidence storage formats focused on the ability to store objects smaller than whole media images.

Phil Turner proposed a format called DEB (Turner 2006; Turner 2005) which included a hierarchical structure to the forensic file. DEBs can contain DEBs. Bradley Schatz extended this format to include extra meta-data about the integrity of the contents.

4.6.2.7 EnCase Logical Evidence Files

EnCase Logical Evidence File (LEF), which was introduced in EnCase version 5 in 2006, extended the existing EnCase Image File formats so that they can store individual files.

Both the EnCase Image file format and the EnCase logical Evidence File have recently been updated with version 2, identified as EWF2-EX01 and EXF2-LX01, where the primary improvement is the addition of compression and string encryption.

4.6.2.8 AFF (AFF, AFD and AFM)

AFF was originally developed by Simson Garfinkel and Basis Technology (Garfinkel et al. 2006). There now appears to be two threads to its development with AFF3 and AFF4 co-existing. AFF4 being authored by Michael Cohen and Bradley Schatz (Cohen et al. 2009). In their paper presented at DFRWS 2009 the authors do refer to distributed evidence but this seems to be the ability to access large evidence containers stored remotely. When data is requested from the remote AFF file, only specific chunks are transferred. In this, it is very efficient but it is up to the administrator to organise the AFF containers in a structure that aligns to the distributed structure of the storage cluster. There is also load redistribution where the most accessible copy of identical AFF files is accessed based on ‘pre-determined distance metrics’.

4.6.2.9 SIP, AIPs and DIPs

In 4.3.7, we considered the OAIS – ISO 14721:2012 system specification. In OAIS, data is submitted as Submission Information Packages (SIPs) and it is converted into Archival Information Packages (AIPs) for storage with the original SIP being lost at the end of the conversion. When data is distributed from the archive, it is converted into another format called a Dissemination Information Package (DIP).

In the previous formats, created specifically for digital forensics, the data file remains unchanged from the time it was created, presumably at the scene of crime by the DEFR, through to the time is used for processing. Arguably additional data files and database entries may be created during processing and these are used to drive the user interface for the investigator.

Woods et al. (2011) explored the use of SIPs, AIPs and DIPs and applied it to the problems of long term archiving of digital evidence.

4.6.2.10 Discussion and Conclusions on current forensic image and storage formats

We can clearly see an evolution in the development of forensic file formats to deal with the digital environment present at the time. Initially single images were preeminent in the investigations but as items like memory sticks and now, cloud storage files have become more common the requirements have changed. Although AFF4 does start to address distributed processing, it is in a form that is rather like a collection of image stores described in 4.3.4 where AccessData have a central storage facility that holds ‘whole’ images. Although there is a ‘use case’ for AFF4 set in a distributed storage environment there does not seem to be any current development in this specific area.

4.6.3 File Sizes, Contents and Fragmentation

In 3.7 we saw that data is now characterised by ‘3V’; Volume, Velocity and Variety and nowhere is this more evident than our analysis task. 10 years ago, we would expect a case to consist of a single hard disk. Now it would most likely be a collection of media from PCs, phones, memory sticks and optical media. Our potential evidence dataset is characterised by diversity.

Successful distributed processing is closely tied to the relationship between the overheads over moving data and processing data. As MapReduce shows, with large datasets it is better not to move the data and to initiate processing locally where the data is stored. However, moving a multitude of small files is very inefficient. This raises the need to gain knowledge of the data ecology upon which we will be processing. As part of this, we have devised a small program that scans a file system and returns statistical data about file numbers, file sizes, file types and file fragmentation. An example print can be found in Appendix B.

We ran the analysis program against a wide variety of data from various Windows based PCs. The resulting data was so diverse as to render any collective statistical analysis effectively useless but there are a number of statements that can be made.

We found that, on the subject of the number of files:

- Windows XP based PC typically had of the order of 100,000 files, Windows 7 based PCs typically had 200,000 files, Windows 8 based PCs typically had 300,000 files, a heavily used desktop PC running Windows 7 sometimes had in excess of 1,000,000 files;

- Certain file types, like picture images, JPGs, PNGs often represented more than 25% of all files by type;

- Often video files, like MP4, represented as much as 40% of file volume;

- A many as 50% of all files were less than 10k. The impact of this is that when small files are received into the cluster, it may be better to keep them on the importing node and process them locally rather than to pass them across to the cluster to process them;

- On PCs running Windows 7and above, with a hard disk as storage, there was very little fragmentation. We believe there might be two reasons for this. Firstly, Microsoft introduced the running of the disk defragmenting program as a standard scheduled task in Windows 7 (Microsoft 2014b). This, of course, should eliminate all fragmentation that would have occurred before the last scheduled running of defragmenter. Microsoft later modified this so that Windows 7 and above attempts to detect Solid State Drives and when detected, does not enable automatic scheduled defragmentation. This modification was done because of the negative effects of excessive writing to SSDs. From a performance perspective, fragmentation has no negative effects on performance as SSDs have zero head latency. Subsequently, when SSDs are used as storage, there was some fragmentation. Secondly, many programs load the entire file contents into RAM for processing. When the user saves their work, the entire file is written as a new file to the media as a stream of data. Then the original file is either deleted or marked as a backup. This is certainly true for word-processing files, graphics files and the like. The exceptions are those where the file was extended over time, for example log files or database files;

- In our sample scan, which counted 325,372 files in total and came from a SSD, only 8,933, less than 3%, were fragmented at all and most of these had five or less fragments to them. These were all log or database files.

We suspect that the low rates of fragmentation, where is does exist on hard disk media where defragmentation has not been run, is because the media is now so large that when a new file is written, there is little trouble in finding a large enough space of contiguous clusters to hold the file. On this assumption, fragmentation would only occur when the partition approaches being full.

4.7 Prioritisation

Although formal triage approaches have been applied within digital forensics numerous times in the last few years, investigators have always exercised a self-taught approach. When faced with finding the ‘data needle’ in the ‘storage haystack’ investigators will, of course, look in the area most likely to hold evidence and act accordingly.

In 2009, Harry Parsonage, at the time a Detective Constable in the Nottinghamshire Constabulary, focused on solutions available to the processing backlog (Parsonage 2009) that he described thus:

> *"At present, in 2009, it is commonplace for digital forensic units to have a backlog, several as long as twelve months."*

His proposals for improving case management by adopting a management triage approach is based on an acceptance that sometimes little or nothing can be done to solve the problem by using better computer processing. In fact, he quotes "work expands so as to fill the time available for its completion" which suggests that he believes if, and when, such additional power is available, the ambition of the investigation will increase correspondingly.

One of the techniques Parsonage proposes is using software that can be configured to focus on specific targets. His example:

> *"The triage software is configured to search for –*
>
> *1. Indecent images using a large hash set of known images.*
>
> *2. Indecent images using fuzzy matching of 25,000 known images.*
>
> *3. Text string search for common terms found in indecent image cases."*

focus on the identification of indecent images as being a key trigger to further examination.

More recently, Shaw and Browne (2013) have reported on their work with Warwickshire Police. They summary the weaknesses of triage as being mainly that of the risk of missing evidence through lack of thoroughness. They propose a system where human inspection, aid by appropriate tools, is conducted on the original media, with safeguards such as a write blocker.

In 2013, Hong et al. (2013) proposed a triage model for digital investigations specifically addressing the issues of large quantities of evidence. They acknowledge the consequence of being overburdened with data is that time-scales are extended to inappropriate durations. Hong proposes that an acquisition strategy can be developed based upon on-going discovery within what they consider a reasonable time-scale to stay on-site.

Hong et al. created a questionnaire that they offered to 97 examiners, of which 58 responded. The questionnaire focused on the practices of respondents when conducting an on-site examination. In particular, it focused on the self-learned data reduction techniques, or short cuts, that the investigators have developed over their time as DEFRs. Half had been DEFRs for more than 4 years of on-site acquisitions.

Their conclusion was that prior knowledge of the investigation subject area allowed them to focus and prioritise data collection and examination. In one section, the questionnaire asked whether the DEFRs believed whether certain activities could be completed on-site, within the "First Golden Hour". The responses to seven key questions are shown in Table 12. The highest response for each activity is highlighted in grey.

On-Site Activity	Always	Quite Possible	Normal	Sometimes/ Partial	Impossible
File Carving	1	3	9	22	23
Restoration of Formatted Media	2	6	11	26	12
Decrypting Disks without co-operation	0	0	4	10	44
Decrypting Files without co-operation	0	1	0	7	50
Finding data hidden by Stenography	0	1	0	12	45
Finding data in files where the extension or signature has been changed	3	7	10	18	20
Recovering data from Corrupt files	0	1	13	26	18
Contents Searching for text	1	8	10	23	16

Table 12 - What can be done on-site? (Hong et al. 2013)

We suspect that had the questionnaire been completed ten years ago the most common selection would have been further to the left, i.e. more possible on-site processing with less data to identify, collect and preserve.

To determine current working practices in more detail, they defined four categories of crime and asked the respondents to express their priority in selecting data files based on their experience of the sources of evidence in these crime types. The categories are shown in Table 13, they are:

	Personal Initiated in a Domestic Setting	Corporate Initiated in a Business setting
Hi-Tech	where an individual engages in a crime that uses digital technology as its methodology, for example hacking initiated from home.	This is typically the 'insider threat' of staff having limited access to a corporate system and exceeding that to gain unauthorised access.
General	where an individual engages in 'traditional' crimes like fraud or counterfeiting but actions them on a digital device.	This may include sexual harassment against another member of staff or spying on confidential documents.

Table 13 - Hong's Crime Scenarios

Given these groups of file types:

- Documents, word processing, spreadsheets;
- Engineering drawing;
- Graphic;
- Voice file;
- Video file;
- Internet history;
- Email;
- Messenger;
- Registry files;
- Event log;
- Executable files;
- Accounting data files;
- Source code files;
- Link files;
- Printer spool file;
- Thumbs.db.

Hong asked what priority the DEFRs would give to each of the file types in each type of crime scenario.

There is no record as to whether the respondents agreed with the relevance or significance of the classification but Hong reports that they chose top 10 sequences as shown in Figure 51:

	Personal	Corporate
Hi-Tech Crimes	1 Internet History	1 Email
	2 Document	2 Documents
	3 Registry	3 Engineering
	4 Event log	4 Internet history
	5 Email	5 Messenger
	6 Executables	6 Graphic
	7 Link files	7 Link files
	8 Engineering	8 Registry
	9 Messenger	9 Event log
	10 Source Code	10 Printer spool file
General Crimes	1 Internet History	1 Document files
	2 Documents	2 Accounting files
	3 Email	3 Email
	4 Messenger	4 Messenger
	5 Video Files	5 Internet History
	6 Link Files	6 Link files
	7 Voice Files	7 Registry Files
	8 Graphic Files	8 Graphic Files
	9 Event Log	9 Event Log
	10 Registry	10 Engineering

Figure 51 - Four Categories of Crime and sources of evidence (Hong et al. 2013)

Horsman (Horsman et al. 2014) introduced the idea of a Primary Relevance Figure (PRF) which uses Bayes' theorem with prior knowledge of previous cases to assign values to the location of potential evidence.

> *"Each location on a system that contains evidence related to an investigation is given an evidence relevance rating (ERR) by the investigating practitioner. The ERR represents the investigator's assessment of the relevance to the case of the evidence found in that location. The ERR is a value between 0.1 (low relevance) and 0.9 (very relevant) in increments of 0.1. For example, for a particular case, evidence found at location C:\Folder might be assigned an ERR of 0.8 to denote that the data found there was highly relevant, that is, the evidence would be strongly relied upon in determining the outcome of the investigation. Research has shown that more fine-grained scales do not provide optimal opinion information during a rating exercise"*

> *"The PRF is the inferred probability that a given location is likely to contain evidence that is relevant to a DT case. As new cases are added to the knowledge base the PRF (i.e., P(E|L)) will change to reflect the new case knowledge"*

Figure 52 shows the changing value of PRF as further cases are assessed.

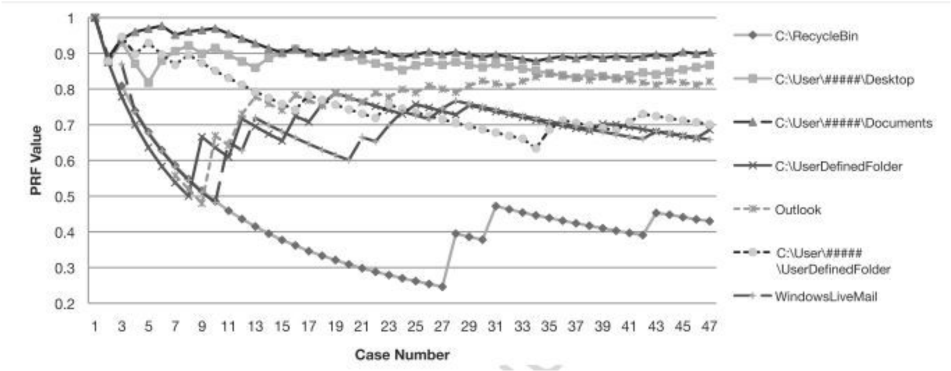


Figure 52 - PRF changing as further knowledge is gained (Horsman et al. 2014)

In an earlier paper, Horsman et al. (2011) addressed the concept of profiling and draws on Rogers (Rogers et al. 2006) earlier observation that despite the "need and want" there is a lack of data generated from digital investigations about criminal behaviour and personalities.

Hong and Horsman's work is a move to the idea that, at least some, investigation work could be automated by an appropriate and relatively simple score system. This might be focused on data-reduction and prioritisation based upon prior knowledge of the characteristics of the crime being investigated.

4.8 Conclusions

We explained the current options within distributed processing and presented previous work in the field that is relevant to our domain and then focused on a specific area of concern; the practice of working with forensic images.

In 4.6.3, we assessed existing image formats and concluded that none was particularly effective in a new environment that did not exist when they were designed. We identified provision of Chain of Evidence as the single worst failing in the existing proposals for distributed solutions and that this has most likely come about by familiarity with the status quo that has existed for 20 years.

In the next chapter, we will form these ideas into design criteria, which we will implement as a prototype when we present FCluster, FClusterfs and our proposal for acquiring data into a distributed system.

5 Designing an Extensible Digital Forensic Investigations Solution

5.1 Introduction

In this chapter section, we review chapters 3 and 4 and develop the framework of a solution. Section 5.2, is intended to read as a representation of a thought process drawing on the issues, criteria and requirements identified in the previous chapters. Having more focus on a solution, section 5.3 presents a list of identified problems and in 5.4 we describe the proposed solution.

5.2 Design Discussion

In chapter 3, we established the historic background and current practice in digital forensics. A lot of this has been adopted and adapted from analogue forensics. One key concept carried over is that of assurance and we see this most notably in the form of a Chain of Evidence used to prove the provenance of evidence collected and subsequently presented in court.

Over the last 15 years, processes have been developed that provide assurance through an effective chain of evidence when processing data for forensics. This is a painstaking procedure of record keeping that does not differ greatly from any other rigorous audit trail. Forensic investigators are already struggling with the volume of data they are required to analyse and it looks as if it will continue. It has been suggested that this backlog of work is having a degrading effect on the legal system. Cutting corners in the legal process is not acceptable. The problem can be summed up as needing to improve processing efficiency while maintaining the quality assurance standards of processing.

In chapter 4, we described several possible solutions. Scaling up, with GPUs and exotic multicore architectures, either often requires complex new programming paradigms or is disproportionately expensive in gaining processing power. Improving processing techniques requires a ‘win’ in many situations, one for each problem and it may be that not every problem has a new elegant solution. This approach could result in a long hard battle and may not yield much gain overall for all the effort. Case triage is reported to have achieved marked success in reducing the case backlog in the last few years. It is, however, open to the criticism that selectively ignoring evidence, or even entire cases, to focus on others based on the, all be it, experienced mind of a professional investigator, holds the risk that critical data will be passed by. Data reduction, in which data is processed to remove irrelevances and highlight connections or repetitions, combined with visualisation is a good candidate but these will surely increase the need for more processing power. Choosing to work to increase processing power in an economic and palatable form is the solution that has most advantage. It would allow bigger algorithms to run in reasonable times. Shorter processing times could be achieved by simply using more hosts.

There have been a few attempts to achieve this but none has had significant success. The common factor in these designs is that they retain the idea of a central file store that holds all the forensic images of the media under investigation. The forensic image, in the basic form of a raw byte-stream or more likely as an EnCase Expert Witness Format file with error checking built-in, has become such a foundation that the profession seems unwilling to give it up. It has attained this status because as it is a facsimile copy of the original and can be checked for integrity by applying a cryptographic hash, it represents completeness and so we gain the assurance of an acquisition job completed and verifiable.

The problem when processing whole images is that when several of these images are held on a central file store, as is the case with AccessData FTK distributed processing, and accessed simultaneously for processing, the connection between the storage and the network switch becomes a bottleneck. We have seen that about 16MB/s can be processed on a PC based on an i7 processor. One storage server can supply enough data for about five host PCs before the Gigabit Ethernet connection limits data supply. Beyond this, there are scaling problems. It doesn’t matter how big the file store, the data can’t be moved to the processors fast enough to occupy any more than about five i7 PCs.

This type of problem, encountered when processing large amounts of data is not unique to digital forensics. A decade ago, when attempting to analyse the contents of large log files it became clear that moving large amounts of data across a network to get it processed was increasing the primary limit on system performance. What was needed was an architecture where the data was already distributed and stored across the network, which would enable data to be processed locally on each storage host node without any, or perhaps minimal, data transfer across the network.

The Hadoop system was designed to solve just this sort of problem. It succeeds because of a combination of a purpose designed processing model, MapReduce, and a purpose designed file system Hadoop File System (HDFS). In an HDFS file system, huge files are broken into chunks; these are typically 64MB with the multitude of chunks stored across the distributed storage facility. Efficient processing of data requires that processing one chunk of data does not require access to data in another chunk, as this would need network access, which is to be avoided if possible. The MapReduce processing model splits processing into tasks and processes just the data stored locally. HDFS is designed to handle a small quantity of very large files that do not require random access to process. Unfortunately, this does not describe the nature of the task during forensic processing of a large image file that does require random access within the image file.

Using the idea of splitting data across hosts, it might be better to acquire digital evidence as individual files and store them across the network hosts but this would mean dropping the idea of a taking a complete forensic image of the media and instead using selective digital evidence containers. This saving on processing time, on its own, may not be justification enough but in addition to this, the practice of taking a forensic image of the whole of the media is coming under attack for three other reasons. Firstly, because a forensic image collects everything on the media, data not relevant to the case is captured as well. There are increasing arguments that the collection of data using whole images is illegal as it is unreasonably wide in its action. Secondly, with the uptake of remote storage systems like Dropbox and Google Drive, there is sometimes no longer direct access to the media and so it is impossible to complete a forensic image. Gradually, the practice of capturing data in the form of forensic images is decreasing and in Digital Evidence Containers is increasing. Thirdly, the increase in media size is making the process of imaging untenably long.

There are however, problems with adopting this approach as a processing model within digital forensics. Collecting data selectively and storing them in digital evidence containers would surely result in slower imaging. This is because the hard disk actuator arm would need to repeatedly seek data across the media rather than run through the sectors in a linear manner. In addition, there would be a considerable task in keeping track of all the individual data files when they are stored on the distributed system.

On the later problem, it seems likely that a database, over and above the file system would provide a solution. There are several ways this could be implemented.

Firstly, it is possible to store entire image files as a Binary Long Object (BLOB). It seems that the largest BLOB allowed in most SQL databases is 2^{32} = 4GB. This is nowhere near enough to storage an entire typical forensic image file. It would however, allow most current files to be stored individually. Under current expectations, only email and video files regularly exceed these sizes. How large individual files may be in the next few years we do not know. It is possible to mimic the clustering of data found in file systems like NTFS and EXT3/4. In this case, individual blocks could be up to 4GB and whole files would be limited only by storage capacity. However, SQL databases take data space allocation upon their own management control. It would be difficult to achieve data locality awareness in processing in the same way as has been so successful with processing in Hadoop.

A more promising approach would be to use the SQL database to mimic just the directory files of a conventional file-system and then hold data within each database record to point to the file containing the data that would be stored in conventional file-system space. In fact, it would be possible to actively place the data file in a specific location, suited to that data and record the link in the SQL database. The SQL database could also host extra meta-data about integrity, evidence source, collation data and time and the like.

The provision of an information store in the form of an SQL database would work but it would mean that every program would need to be aware of the database. This would mean amending existing software and incorporating this functionality in the design of any future software. If would be far better if this was completely transparent to the application software. In fact, it would be better if this functionality could be written into the operating system, as application programs would be unable to avoid the control. Writing a new operating system specifically for digital forensics is a practical option. Luckily, there is an intermediate solution. This could be written as a File System in User space. We reviewed several FUSE file-systems and this does seem very similar to a combination of the features found in existing FUSE file-systems. We should then think of a custom file-system that controls all access to evidential data.

The downside of this is that we can still expect this to incur a management overhead. The additional time spent managing the data as it moves around the network needs to be regained in the advantages from embarrassingly parallel processing when it is in place. Where possible, processing should be undertaken on data held locally.

On the issue of slower imaging using selective DEC techniques, the solution might be to introduce some form of prioritisation. Roussev’s LOTA imaging strives to make whole data files available as soon as the linear imaging process passes the last cluster in which data for that particular file has been held. Longer overall imaging time may be acceptable if highly valued data is available much sooner than by previous techniques. A file based, evidence container, collection method that targets files that have been given a high value for their potential evidence value should offset the latency we can expect from the costs of the management of data across a distributed system.

Figure 53 is a representation of the current practice. With larger media needing investigation, the first two stages, where no profitable work is actioned, are taking longer. It is only in the final stage that the investigator actually gets to conduct meaningful investigation, producing evidence to be used in the case.

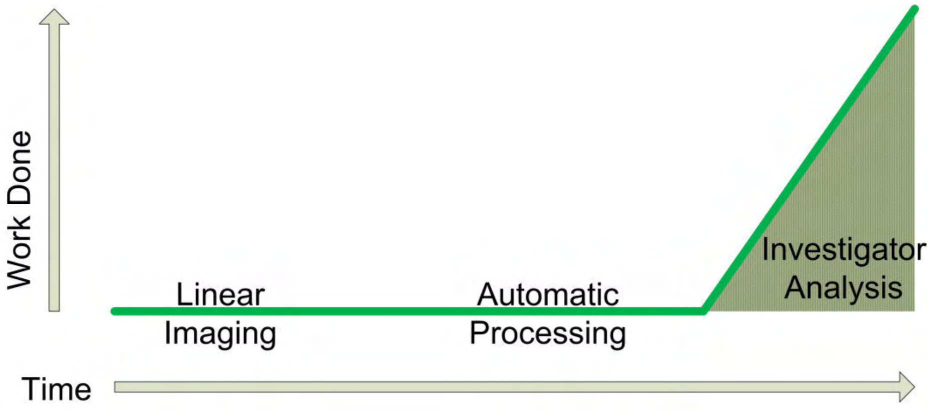


Figure 53 - Productive Investigation Time with Conventional Processing

Figure 54 shows a representation of what we hope to achieve. By extracting data from the source and delivering it to a distributed processing architecture, we hope to draw forward the productive time of the investigator. Prioritisation will mean that the data considered by the investigator to be of greater likelihood of containing evidence will be drawn forward in the queue and so can be considered much earlier.

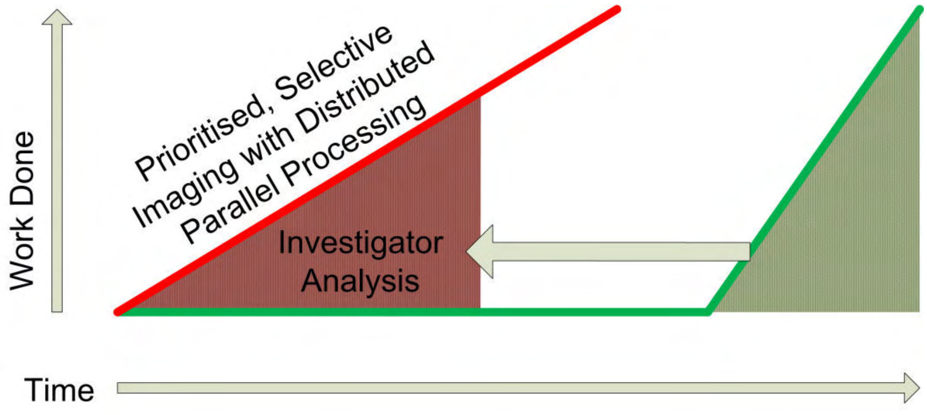

Work Done
Prioritised, Selective Imaging with Distributed Parallel Processing
Investigator Analysis
Time

Figure 54 - Improved Investigation Time with Distributed Processing

5.3 Identified Problems (Design Objectives)

The previous work has resulted in a design requirement that identifies 18 problems. These are listed in detail in Table A1 in the Appendixes. They are categorised and associated with a summary description of the problem.

They are briefly:

1. The blocking nature of Linear Imaging;
2. Processing and data transfer balancing;
3. Focusing processing on potentially high yield source data;
4. Complex, time consuming tasks, can block processing flow;
5. The need for meta data from the original source to be available throughout the process;
6. Access to raw data needs strict control;
7. Legacy software should not be excluded;
8. From the point of view of the programmer, the system should be as close to existing paradigms as possible;
9. The difficulty in handling large amounts of data from diverse sources;
10. Data transmission needs to be secured;
11. Data disposal needs to be monitored and secure;
12. Audit must be transparent;
13. Data transfer and replication must be transparent;
14. Data must be read-only;
15. Data integrity must be frequent and transparent;

16. Validation of processing results by replication of process;
17. All source data needs a unique identification;
18. Automation should be maximised.

5.4 The Solution Outline and overview

Therefore, in summary, our design has come to an SQL database based FUSE file-system. The database will hold meta-data but not the data itself. This will be linked via reference to a file stored in the host operating system, which is ext4 on Linux.

The meta-data, stored in the database will be sufficient to assure the quality of data handling on the distributed processing platform. Access to data will not be possible without reference to the FUSE file system.

Data will be acquired as selective digital evidence containers that must have sufficient provenance information and integrity meta-data included in its contents.

Acquisition will not be in a conventional linear action but by using the file-system metadata, the directory, to prioritise collection based upon a scorecard system with values set by the investigator.

Because evidence is acquired in discrete digital evidence containers, they can be dispatched to a processing facility immediately after they are created. They will be taken into the cluster by the middleware that organises load balancing and distribution. Integrity checks are run as a routine act. All this is controlled by the middleware.

Processing can be initiated as soon as each one is received and verified.

The previous sections lead to a solution that can be described as:

- Middleware
    - Having an operational model that is implemented as a middleware that runs on an existing operating system. This leverages all the build assurance of a tried and tested infrastructure. In which all data movement, unpacking and verification will be implemented by a series of operating system scripts;
- File System
    - Is based around a custom file system in which file meta-data data is stored within a well-established SQL database. All the existing facilities of this SQL database will be inherited by the forensic system;
    - one in which access to data for processing will be only via a custom file system implemented using FUSE technology;

    - one in which evidential data will be stored as whole data files, not as partial chunks or blocks;
- Processing
    - one in which data processing will be actioned locally where possible but can be actioned remotely via a network connection;
- Organisation
    - one in which the system can be divided into zones of assurance;
- Security
    - one in which data will be moved and stored in an encrypted form
    - one that implements fine-grained access control.

5.5 Conclusions

In this chapter, we have constructed a design discussion that has led to an outline design solution. In the next chapter, we will describe this solution, in detail.

6 A Middleware Solution for Forensic processing on a Distributed System

This chapter introduces a design for a solution to the problems identified in chapters 3 and 4.

Section 6.2, presents an overview of the whole FCluster system describing the architecture together with a brief description of the components.

Section 6.3, describes FCluster in terms of information assurance and the passage of data through those zones.

During the design process, it became clear that although it would certainly be desirable to adopt a digital evidence container that has an active design and development community behind it, the use of the most likely candidate AFF4 caused too many complexities in our prototype. Instead, we substitute a greatly simpler design, described in section 6.5, that embodies the principles of the DEC requirement with few of the sophistications.

FCluster comprises of a new imaging process we have chosen to call Jigsaw Imaging, a data processing priority algorithm and the FClusterfs File system. These are introduced in sections 6.4, 6.6 and 6.7 respectively.

This chapter finishes with some conclusions, in section 6.8.

6.1 Introduction

FCluster is a peer-to-peer middleware for a network of heterogeneous host computers. FCluster has controls for data ingestion, movement and availability in a computer cluster in such a way that the data's integrity and authenticity is assured throughout its existence. We take an holistic approach to the forensic process by extending this assurance from acquisition right through to processing. FCluster includes a design for a DEC with a simple structure. FCluster is a vehicle upon which application programs can be run; it is not an application program. It is language agnostic and so does not require application programs to be written in a specific way or language.

Table A1 in the appendixes shows each facet of the solution against its corresponding problem.

The movement of data through FCluster is achieved by several control programs that interlock in such a way that subsequent control programs cannot process data that does not fulfil the requirements of the successful completion of the previous control program.

They form a series of four interlocking assurance zones; shown in Figure 55.

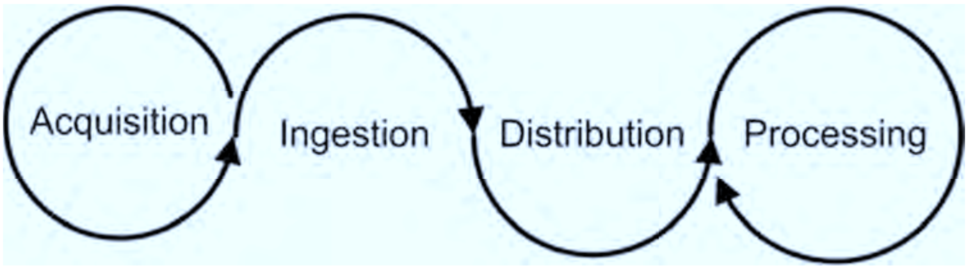


Figure 55 - FCluster Assurance Zones

FCluster is based on a new FUSE file system, FClusterfs that mimics the functionality of a regular file system but writ large. FClusterfs is described, in detail, in section 6.3.

The current assurance zone for a particular item of evidence can be determined by the presence or absence of data in the MySQL database behind FClusterfs.

6.2 Architecture

In this section, we present a broad overview of the components that make up FCluster.

6.2.1 Hardware topology

FCluster is intended to run on very conventional hardware often called Mass Market Commodity off the Shelf Processors (M^{2}COT). This is the same approach as Hadoop and many other clusters. FCluster can run across local and wide area networks. While all the data transfer is encrypted via a VPN, we accept there are legal issues associated with the geographic storage local of the data that would be analysed within FCluster. This issue is outside of the scope of this project; see section 1.3.3 .

6.2.2 Software Architecture – Middleware

FCluster is implemented as a middleware that sits on top of an existing operating system; see Figure 2 - System layers. The prototype is implemented on Ubuntu 12.04 but we see no reason that this cannot be extended to Linux in general, OSx and Windows.

6.2.3 Host Roles and Servers

FCluster comprises of a number of servers. A single host may have several servers and thus fulfil several roles. Each host computer in the cluster might fulfil all of the roles listed below, but it is likely that most hosts will be allocated just three or four roles.

6.2.3.1 Acquisition Authority Server

This manages the cryptographic keys used to authorise imaging. These keys are stored in the MySQL database at the heart of FClusterfs. These keys are issued to the imaging devices.

6.2.3.2 Imaging Device

These portable devices read accept the cryptographic keys created by the Acquisition Authority servers and use them to encrypt data from the original media and create the directory metadata Digital Information Containers (DECs), file data DECs and Image files. Imaging devices are considered part of the assured system of FCluster. They need to be uniquely identified and the cryptographic keys created by the Acquisition Authority Server are uniquely allocated to a specific imaging device. The Acquisition process, which we will still call imaging although the primary output is not an image but a collection of digital evidence containers, encrypts all the data collected.

6.2.3.3 FClusterfs file-system metadata storage server

The heart of FCluster is a multi-featured File System in User Space (FUSE) filesystem. FClusterfs is a hybrid amalgamation of the existing FUSE file systems described in 4.5 with some original extensions. None of these existing FUSE file-systems, in themselves, provides all the features needed to satisfy our requirements. FClusterfs can be thought of as a superset of, in particular, MYSQLfs with considerable amendments to their code to enable extra functionality.

6.2.3.4 DEC Ingestor/Importer

This firstly identifies new evidence DECs that were created by an Imaging device, and have been placed in an input staging area. They are then validated to assure that they are expected. This triggers the start of ingestion.

6.2.3.5 Load Balancer

The load balancer takes DECs from the DEC ingestor and then chooses which storage/processing host should hold the primary copy of the data based on the processing server's capacity and workload.

6.2.3.6 Replicator Service

The replicator's primary task is to see that there are enough copies of the DECs to ensure redundancy. Its primary sub task is to identify and place the primary copy of the DEC on its allocated storage server. Subsequently, its subordinate tasks are to ensure the secondary and tertiary copies are in place. The Replicator service always 'pulls' data. In this way, each server is responsible for obtaining and managing its own data, thus distributing the task.

This service also verifies that the data on the storage server is still valid by routinely performing cryptographic hash calculations and comparing these to those stored in FClusterfs' MySQL database.

6.2.3.7 Data Storage server

This server actually holds the data. All data is stored in the native file system used by the operating system. For example, on a storage server running Linux this would most likely be ext3/4. Data is stored within the native file system is encrypted and needs to be decrypted before it can be interpreted.

6.2.3.8 Processing Host

This role controls the actual processing of data. To avoid any unnecessary data transfer, this would usually be combined with the data storage role.

6.2.4 Scale and scalability

FCluster is intended to provide distributed processing to regional forensics labs. At the start of the project, in 2011, we arbitrarily chose the budget of £30,000 as being realistic and that this would buy about 100 host PCs. Each might have 16GB RAM and 3 TB of hard disk storage and be linked together with a gigabit Ethernet network on a single 48-port switch. FCluster is intended to be implemented on small clusters. It is not certain where scaling problems would occur but these are outside the scope of the project; see section 1.3.3 .

6.2.5 Peer to Peer and multi file-system

MYSQLfs (Brancatelli 2014) is presented by its authors as a single database solution in which one database is hosted on a MySQL server to hold the metadata and data of a single file-system which is mounted on the host file system.

In FClusterfs, we extend this to enable any single MySQL server to hold any number of databases that can each hold any number of file system volumes.

Within the limits prescribed by the access control system of MySQL, a user can access any MySQL server and then any database within that and any file-system within that. In reality, it is unlikely that every host on a cluster would provide a MySQL database service for FClusterfs.

6.3 FCluster

In this section, we will describe FCluster in much more detail.

6.3.1 Introduction

FCluster can be understood as a collection of components that interrelate and depend on each other for the assured transit of data through the system. FClusterfs is the file system that handles all the data interactions during this transit.

6.3.2 An Overview of Data transfer through FCluster

Having established the component parts of FCluster from a high-level view, it is now possible to demonstrate its operation by following data as it is gathered and passed into the system. Figure 56 show this as a series of data movements controlled by the control daemon programs.

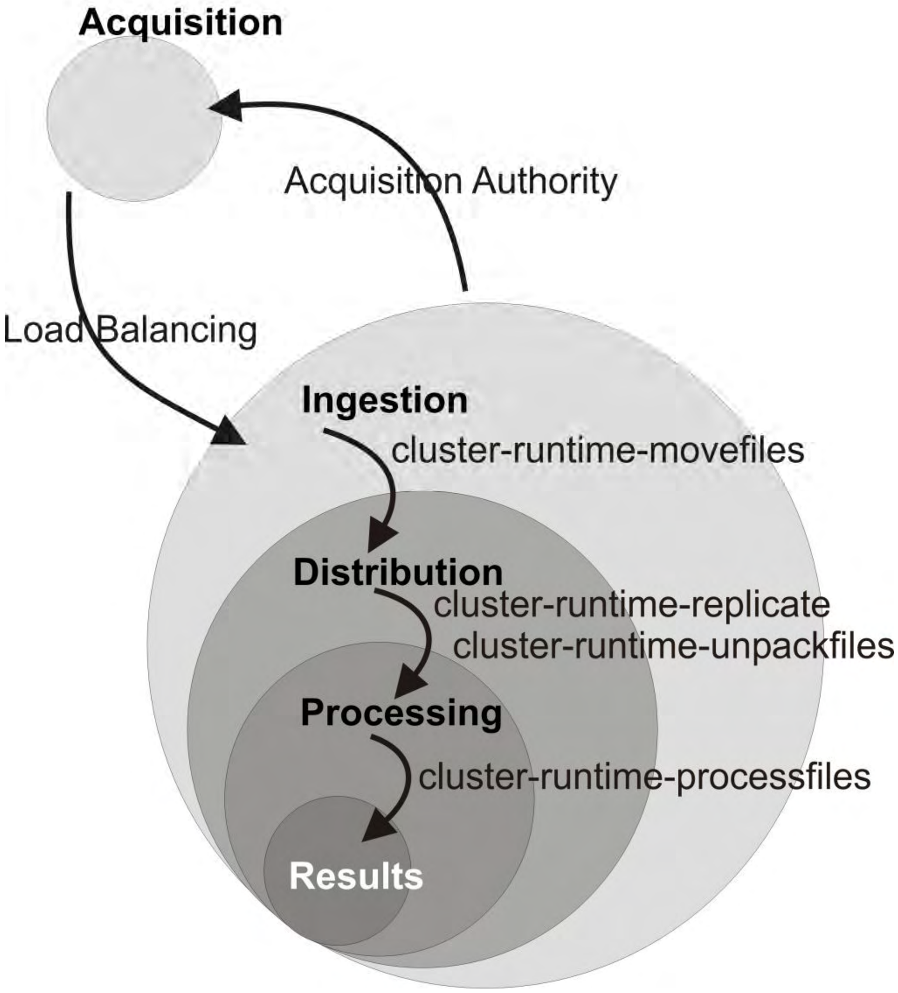

Acquisition
Acquisition Authority
Load Balancing
Ingestion
cluster-runtime-movefiles
Distribution
cluster-runtime-replicate
cluster-runtime-unpackfiles
Processing
cluster-runtime-processfiles
Results

Figure 56 - Data Movement through the Zones

The whole process is initiated by an administrator creating an Acquisition Authority - “Authority to Image”. This is in the form of a file that is created on the cluster and contains a key text string to be used for encryption. This is then passed to the imaging device.

The imaging device uses the data contained in the authorisation file to encrypt the data as it is acquired and stored as DECs.

The imaging process has four deliverables:

1 a single DEC containing directory metadata;

2 a collection of DECs, one each for each file that falls into a ‘high value’ criteria set by the image acquirer;

3 a conventional ‘forensic image’, for reference and later extraction of further data;

4 and DECs for unallocated disk space.

The imaging process is explained in detail in section 6.4.

The selection of files to be packaged as DECs is based on a user determined priority, which is explained in section 6.5, and subsequently only file types expected to have a higher likelihood of containing evidence depending on the case type are collected at this stage.

The ingestion of data into FCluster is initiated as a part of load balancing when the directory metadata DEC, containing the data defining the file system directory, is imported into the MySQL database at the heart of FClusterfs. At this stage, a directory skeleton will exist within FClusterfs but no data is available for analysis.

The file data, in the form of a multitude of DECs, is imported as it becomes available. If the Acquisition device and the cluster are connected, data DECs are ingested almost as soon as they are created. When the Acquisition is remote from the analysis cluster the DECs are stored and forwarded later. The arrival of the data DECs on the cluster starts a process of ‘filling out’ the evidence file system with data associated with each directory entry.

The allocation of DECs to storage servers is based upon calibration benchmarking that is undertaken locally on the storage/processing servers and reported to the MySQL database. This allocation information is saved within the MySQL database behind FClusterfs.

The cluster-runtime-movefiles daemon is constantly scanning the FClusterfs MySQL database to find files allocated to storage local to the running daemon. When it learns of one, it copies it onto its local storage.

The cluster-runtime-unpackfiles daemon is constantly scanning to find DECs as they arrive at the storage/processing servers. When a DEC arrives on its storage host, it is unpacked and its contents are verified against the data already held in the FClusterfs MySQL database. Only if it is proven valid is it then accepted and made available via the distributed file system, FClusterfs. Upon approval at its storage location, another daemon, cluster-runtime-processfiles, initiates a defined list of tasks and automatic processing is conducted, for example generating text indexing or thumb-nailing images.

To provide integrity, redundancy and secondary load balancing, a replication daemon, cluster-runtime-replicate, firstly ensures constant and routine validation of data by applying a

SHA 1 checksum calculation to each file and comparing the result with the figure created at acquisition time and stored in the MySQL database. It then ensures that there are multiple copies of the data, normally three, held on separate hosts within the cluster.

The DECs created at image time will, most likely, have captured only part of the evidence. Subsequently a ‘Bag it on demand’ system can trigger an on-the-fly acquisition of data that was initially deemed of secondary interest within the image once it has been completed and is available to the cluster. This data is validated and placed in the same assured manner as the rest of the system.

How FCluster is configured as a network system is up to the administrator but it can form a local or wide area network. The prototype successfully uses a VPN to connect the nodes and we have extended it to use nodes on Amazon Web Services. Whenever data is transferred between nodes is it always in an encrypted form and so can be considered safe in a technical sense but this may not be acceptable on principle within a legal environment. The primary objective, and the core of any speed improvement, is that processing takes place locally on the datanode holding the data. In a similar way to the use of SHA1s to identify ‘Bad’ files, the system can be used without the actual files being accessed. Results are transferred across the network but not normally the data.

6.3.3 Various Perspectives on FCluster

6.3.3.1 By Assurance Zone

The Assurance zones Figure 57 - FCluster Assurance Zones, first seen on page 141, shows a very high-level view of the separation into zones.

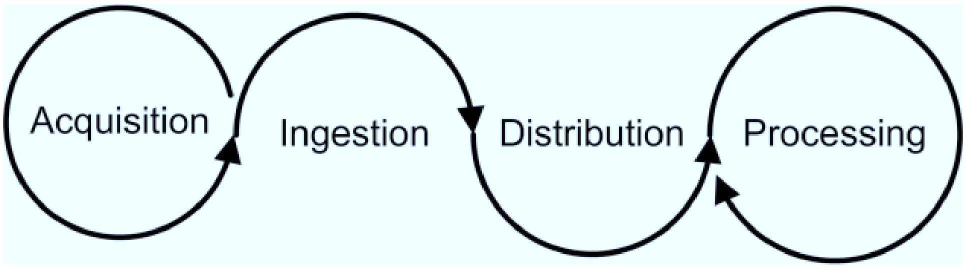


Figure 57 - FCluster Assurance Zones

Figure 58 - Assurance Zones as Rings, expresses this as a series of concentric rings. The progress of data through the assurance rings depends on the successful transition through the previous ring. Ultimately resulting in the processing of data that is assured to be correct in terms of is provenance and integrity within the system.

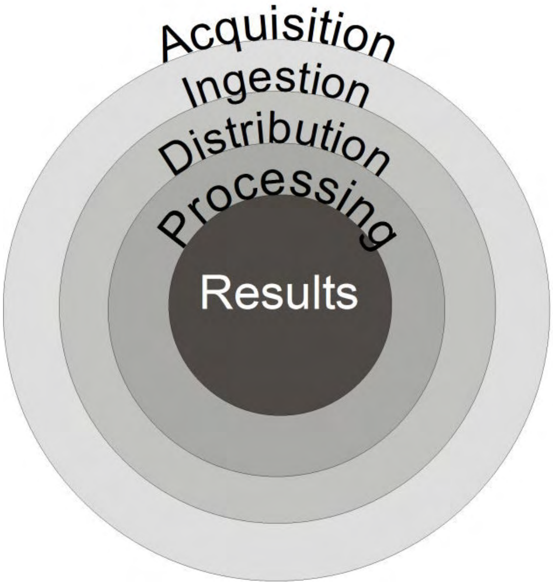


Figure 58 - Assurance Zones as Rings

The properties or attributes necessary to establish assurance of data are all stored within the MySQL database as an integral part of the FClusterfs file system. In the processing zone, there is no other method of accessing data other than through FClusterfs.

6.3.3.2 As a schematic with FClusterfs at its core

Figure 59 - The four Zones of Assurance, shows the movement of data in detail. FClusterfs is the constant control throughout the later three assurance zones.

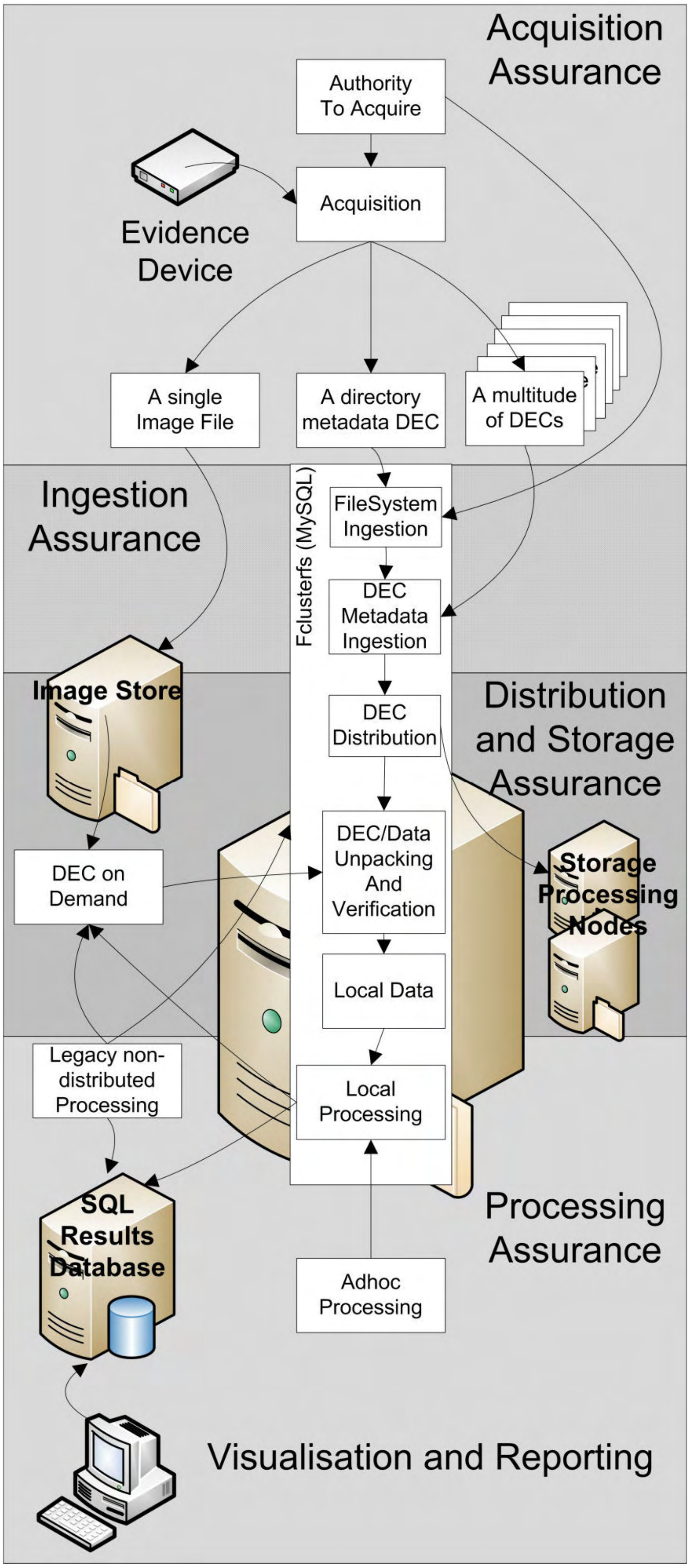


Figure 59 - The four Zones of Assurance

6.3.3.3 *Mapping the Assurance zones to the FCluster system schematic*

Figure 60 shows the assurance zones mapped against the system schematic.

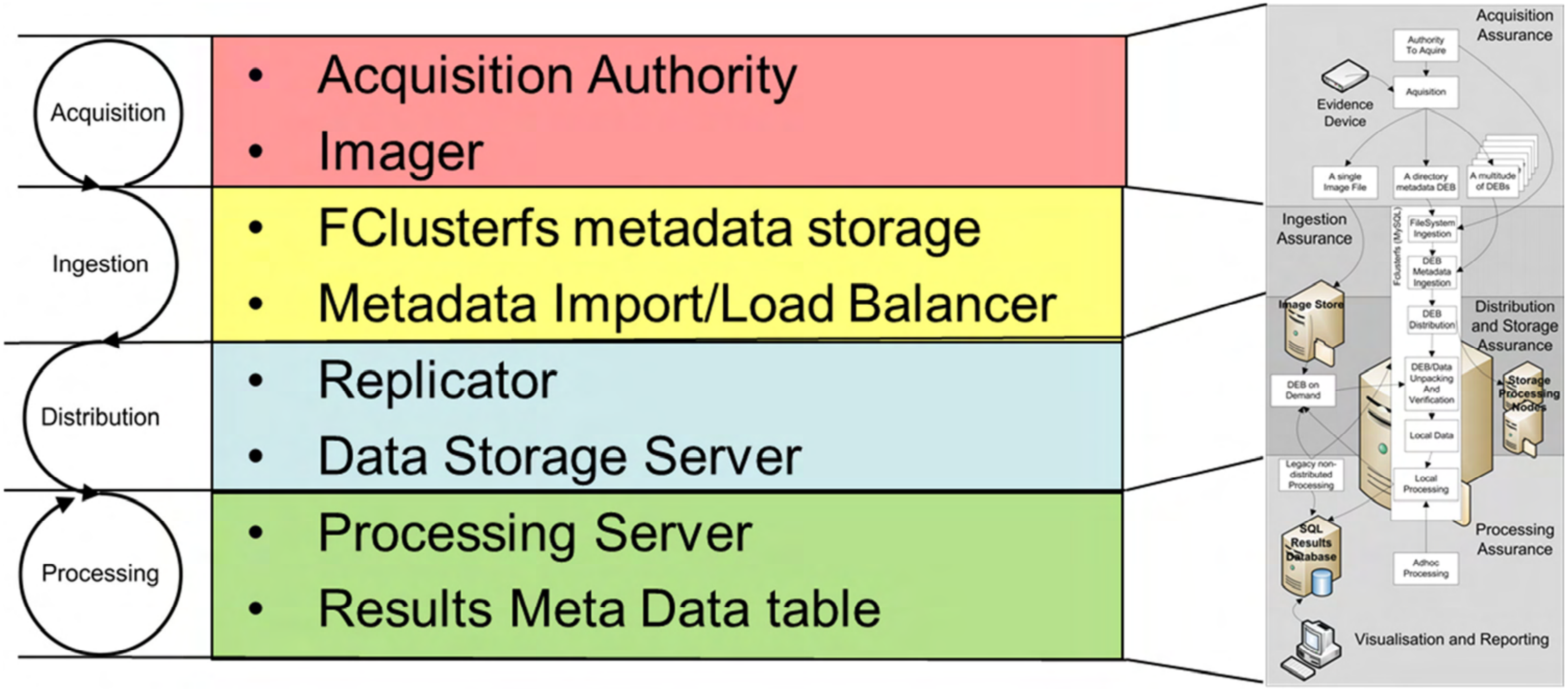
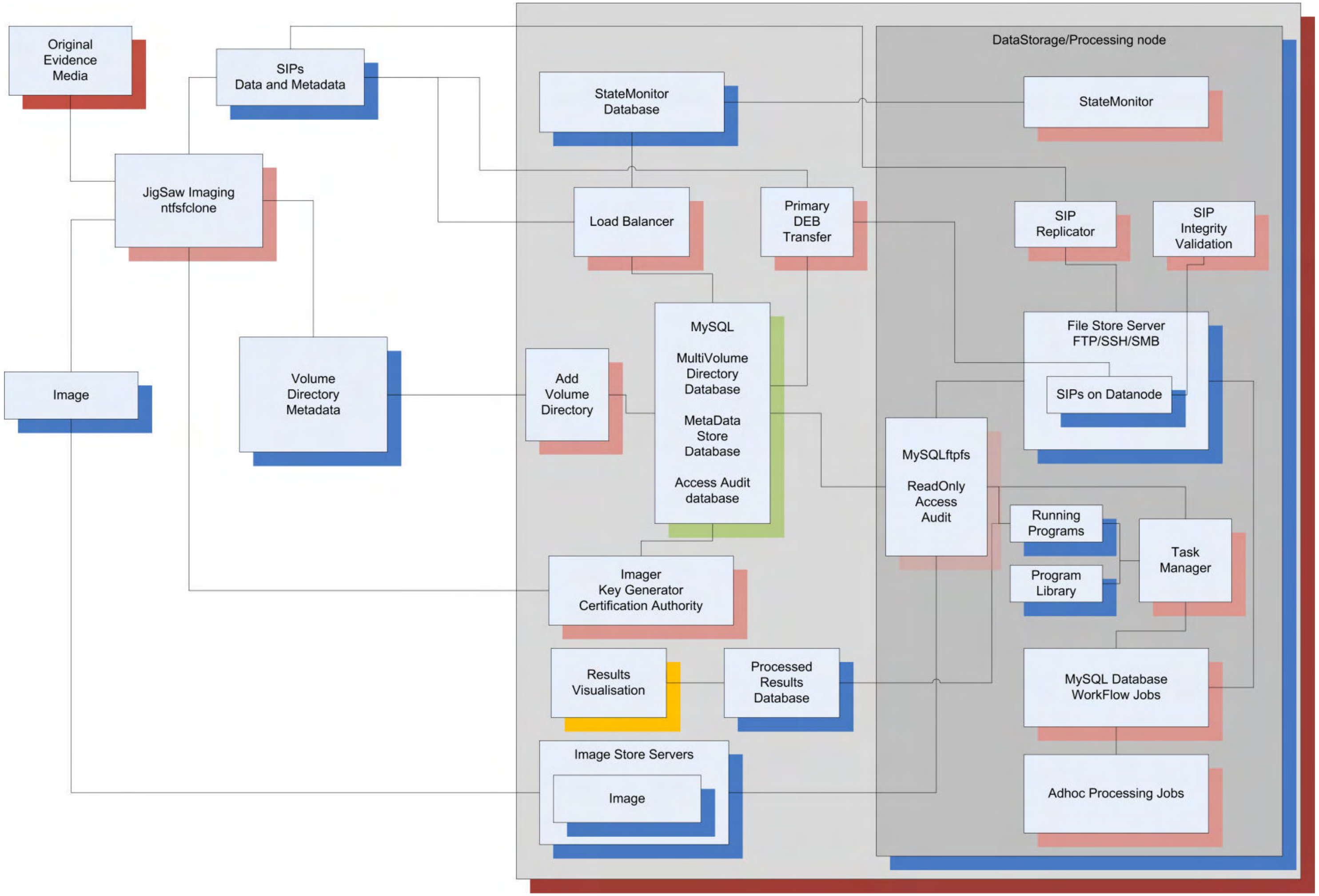


Figure 60 – Mapping FCluster Assurance Zones

6.3.3.4 *FCluster system schematic*

Figure 61 - FCluster System Schematic shows the components and their interrelations.



Legend

Pink: Program

Blue: Data

Green: MySQL

Figure 61 - FCluster System Schematic

6.3.4 Assurance in more detail

6.3.4.1 Acquisition Assurance in Detail

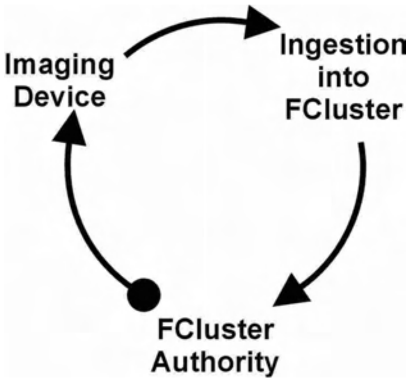


Figure 62 - Acquisition Assurance Loop

The first assurance zone in FCluster is controlled by a “Loop of Authority and Acknowledgement” in which authority is granted to an imaging device to take an image; Figure 62 - Acquisition Assurance Loop. Consequently, FCluster only accepts data that was gathered with an authority issued by FCluster.

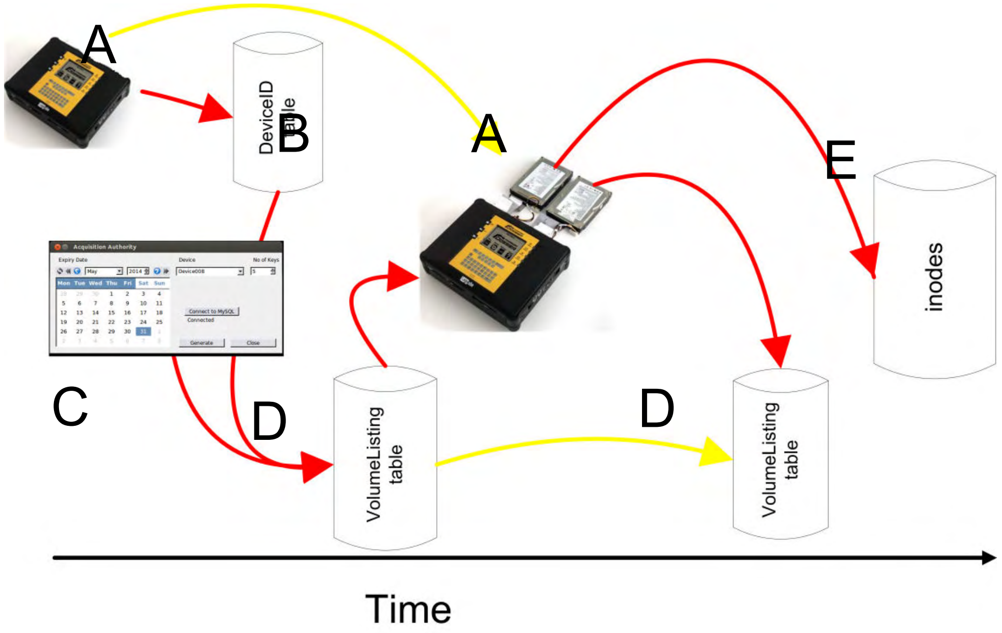
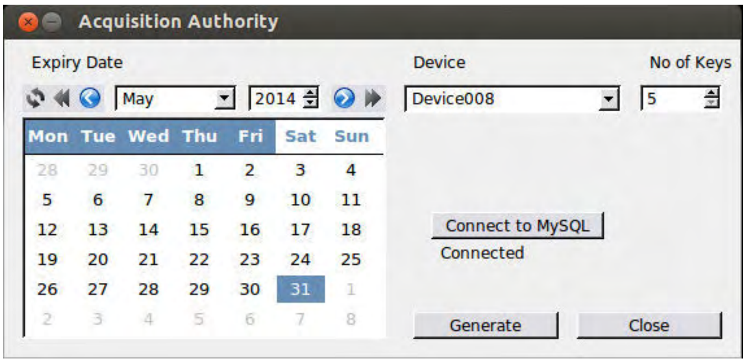


Figure 63 - Imaging Authority over Time

Figure 63, shows the passage of data, in red, and the movement of hardware, in yellow, set against the passing of time. First, the existence of the imaging device “A”, is registered with the FCluster DeviceID table in the FClusterfs database “B”. Then the Acquisition Authority program “C”, shown in more detail in Figure 64, is used to create a random key to be used in encryption. This key, specifically allocated to device “A”, is stored in the VolumeListing table in the MySQLfs database “D”. When device “A” is used for acquisition, is used the key generated at “C” and stored in “D”. FCluster will only accept input if it comes from device “A” and uses the specific key. If it is acceptable, the resulting data is added into the inodes table "E"



Acquisition Authority

Expiry Date

May 2014

Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Device: Device008

No of Keys: 5

Connected

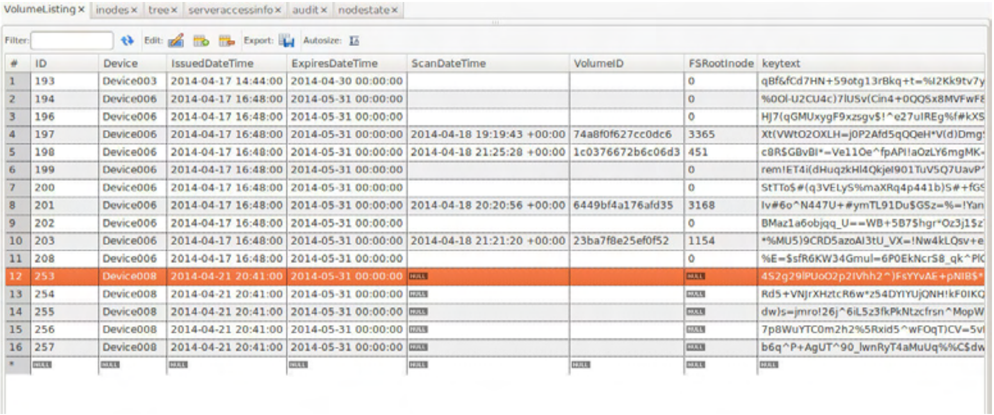
Figure 64- Acquisition Authority Dialog Box

The key information is sent to the imaging device where it is used to encrypt the data within the DECs. The meta-data DEC, who contents are shown in Figure 67, is ingested in to the Cluster where it is matched with the database record of its creation, an example is shown in Figure 65. The directory structure, stored as data within the meta-data DEC, is ingested as added to the FClusterfs database.

The Administrator chooses options within the dialog box to control the contents of the authority file.

- Device Identification: Authority is granted to, and only to, one of the devices known to the system. The details of which are held in the DeviceInfo table;
- Expiry Date: As part of the assurance control, the Authority can expire. This authority will expire on a date set by the Administrator;
- Number of Keys to Generate: The Authority is given to create one or more keys depending on the assessment of the Administrator.

The 'Authority to Image' file contains a reference number and a randomly generated key that will be used at acquisition time to encrypt the data stored in the DECs. The reference number and key are recorded as a new record in the VolumeListing table in the FClusterfs database. Multiple keys can be created and issued to multiple imaging devices to form a 'stock of authorities' to be used over a period of time, the keys have an 'expiry date' associated with them as an added control. The reference number under which the cryptographic key was recorded is located in the VolumeListing table.



#	ID	Device	IssuedDateTime	ExpiresDateTime	ScanDateTime	VolumeID	FSRootInode	keytext
1	193	Device003	2014-04-17 14:44:00	2014-04-30 00:00:00			0	qBf&fCd7HN+59otg13rBkq+t=%I2Kk9tv7y
2	194	Device006	2014-04-17 16:48:00	2014-05-31 00:00:00			0	%0Ol-U2CU4c)7lUSv(Cin4+0QQSx8MVFwF8
3	196	Device006	2014-04-17 16:48:00	2014-05-31 00:00:00			0	HJ7(qGMUxygF9xzsgv$!^e27uIREg%f#kXS
4	197	Device006	2014-04-17 16:48:00	2014-05-31 00:00:00	2014-04-18 19:19:43 +00:00	74a8f0f627cc0dc6	3365	Xt(VWtO2OXLH=j0P2Afd5qQQeH*V(d)Dmg
5	198	Device006	2014-04-17 16:48:00	2014-05-31 00:00:00	2014-04-18 21:25:28 +00:00	1c0376672b6c06d3	451	c8R$GBvBI*=Ve11Oe^fpAPI!aOzLY6mgMK-
6	199	Device006	2014-04-17 16:48:00	2014-05-31 00:00:00			0	rem!ET4i(dHuqzkHl4QkjeI901TuV5Q7UavP
7	200	Device006	2014-04-17 16:48:00	2014-05-31 00:00:00			0	StTTo$#(q3VELyS%maXRq4p441b)S#+fGS
8	201	Device006	2014-04-17 16:48:00	2014-05-31 00:00:00	2014-04-18 20:20:56 +00:00	6449bf4a176afd35	3168	Iv#6o^N447U+#ymTL91Du$GSz=%=!Yan
9	202	Device006	2014-04-17 16:48:00	2014-05-31 00:00:00			0	BMaz1a6objqq_U==WB+5B7$hgr*Oz3j1$z
10	203	Device006	2014-04-17 16:48:00	2014-05-31 00:00:00	2014-04-18 21:21:20 +00:00	23ba7f8e25ef0f52	1154	*%MU5)9CRD5azoAI3tU_VX=!Nw4kLQsv+e
11	208	Device006	2014-04-17 16:48:00	2014-05-31 00:00:00			0	%E=$sfR6KW34Gmul=6P0EkNcrS8_qk^PlC
12	253	Device008	2014-04-21 20:41:00	2014-05-31 00:00:00	NULL		NULL	4S2q29lPUoO2p2IVhh2^)FsYYvAE+pNIB$*
13	254	Device008	2014-04-21 20:41:00	2014-05-31 00:00:00	NULL		NULL	Rd5+VNJrXHztcR6w*z54DYIYUjQNH!kF0IKQ
14	255	Device008	2014-04-21 20:41:00	2014-05-31 00:00:00	NULL		NULL	dw)s=jmro!26j^6iL5z3fkPkNtzcfrsn^MopW
15	256	Device008	2014-04-21 20:41:00	2014-05-31 00:00:00	NULL		NULL	7p8WuYTC0m2h2%5Rxid5^wFOqT)CV=5vl
16	257	Device008	2014-04-21 20:41:00	2014-05-31 00:00:00	NULL		NULL	b6q^P+AgUT^90_lwnRyT4aMuUq%%C$dw
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 65 - VolumeListing Table Sample Contents at 20th May 2014

Figure 65, shows the typical contents of the VolumeListing table. Record 12 contains the 253rd authority granted by the system. The authority is granted to ‘Device008’, and was created on the 21st April 2014 at 20:41 and will expire on the 31st May 2014, in 10 days’ time. The scan may have taken place but the data has not yet been returned as evident from the absence of data in the ScanDateTime, VolumeID and FSRootnode fields. Record one is an example of an authority that will have expired on the 20th May 2014. Record 4 is an example of a scan that is now complete as there is evidence that it has been drawn into FCluster by the presence of data in the ScanDateTime, VolumeID and FSRootnode fields.

We have devised a novel imaging process that provides assurance while also fully exploiting the concurrent nature of distributed processing. We will present Jigsaw Imaging, in detail, in section 6.4.2 . For now, it will suffice to say that the imaging process has four outputs.

1 a single DEC containing directory metadata;

2 a collection of DECs, one each for each file that falls into a ‘high value’ criteria set by the image acquirer;

3 a conventional ‘raw forensic image’, for reference and later extraction of further data.

4 multiple DECs containing the data from the Unallocated file-space.

6.3.4.2 Ingestion assurance

The ingestion section of the zones schematic is shown in Figure 66.

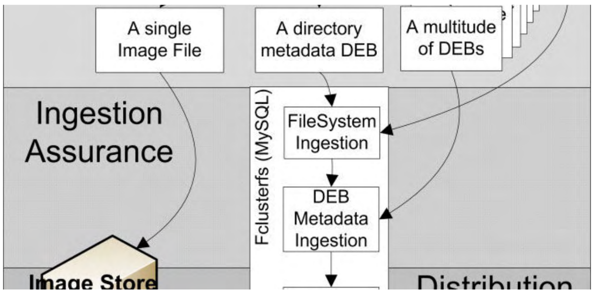


Figure 66 - Import/Ingestion Zone

The DEC containing file-system directory metadata is the first to be imported back into FCluster. The typical contents of the file-system metadata file can be seen in Figure 67 and the controlling dialog box can be seen in Figure 68.

```
<keytext>-NbtMSejN$c&LO^URH9</>
<scandatetime>2014-08-07  0:35:28 +00:00</>
<filesystemSerialNo>14A51B74C91B741C</>
[      4 -rwxrwxrwx        2560 May 11  2011]  ./$AttrDef
[      8 -rwxrwxrwx           0 May 11  2011]  ./$BadClus
[      6 -rwxrwxrwx       32736 May 11  2011]  ./$Bitmap
[      7 -rwxrwxrwx        8192 May 11  2011]  ./$Boot
[     11 -rwxrwxrwx           0 May 11  2011]  ./Extend
[      2 -rwxrwxrwx     5361664 May 11  2011]  ./$LogFile
[      0 -rwxrwxrwx     2127872 May 11  2011]  ./$MFT
[      1 -rwxrwxrwx        4096 May 11  2011]  ./$MFTMirr
[      9 -rwxrwxrwx           0 May 11  2011]  ./$Secure
[     10 -rwxrwxrwx      131072 May 11  2011]  ./$UpCase
[      3 -rwxrwxrwx           0 May 11  2011]  ./$Volume
[   2044 drwxrwxrwx        4096 May 11  2011]  ./Acer
[   2027 -rwxrwxrwx       76800 Nov 30  2008]  ./A Grid GPS part 2.doc
[   2028 -rwxrwxrwx       11805 Nov 30  2008]  ./A Grid GPS.pdf
..
..
[   1942 -rwxrwxrwx        6225 Aug  6  2008]  ./VideoViewer/guard.jpg
..
..
[   2025 drwxrwxrwx           0 May 14  2009]  ./Winhex
[   2026 -rwxrwxrwx     1246735 Jul  3  2008]  ./Winhex/winhex.zip
[   2042 -rwxrwxrwx     1373751 Jun 23  2009]  ./wrar39b3.exe
46 directories, 1967 files
```

Figure 67 - The Contents of the Image File-System Metadata DEC

The design for the simple DEC is given in more detail in section 6.5.

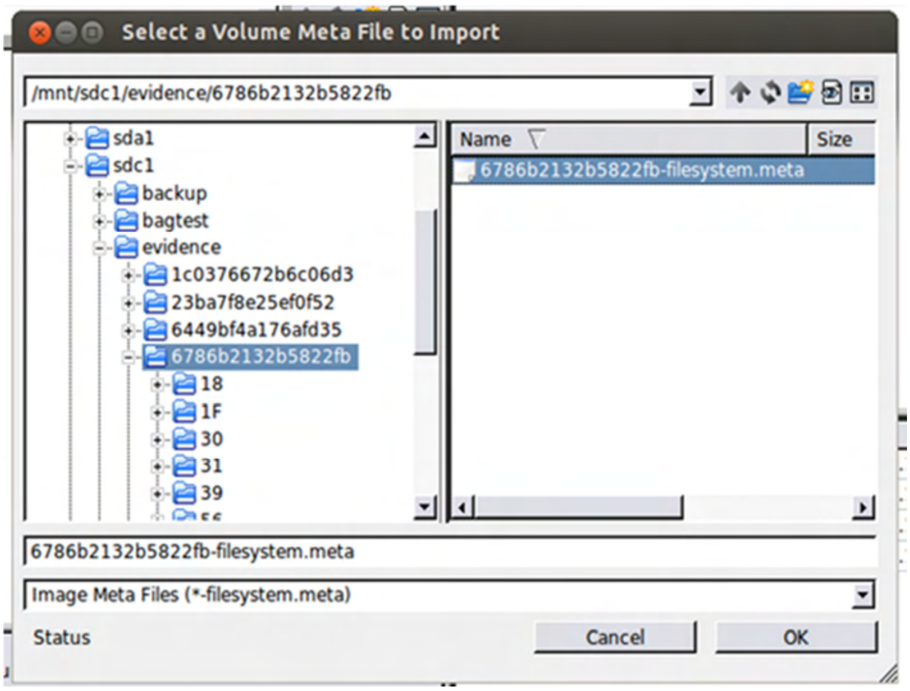

Select a Volume Meta File to Import
/mnt/sdc1/evidence/6786b2132b5822fb
sda1
sdc1
backup
bagtest
evidence
1c0376672b6c06d3
23ba7f8e25ef0f52
6449bf4a176afd35
6786b2132b5822fb
18
1F
30
31
39
Name
Size
6786b2132b5822fb-filesystem.meta
6786b2132b5822fb-filesystem.meta
Image Meta Files (*-filesystem.meta)
Status
Cancel
OK

Figure 68 - File-System Volume Metadata import

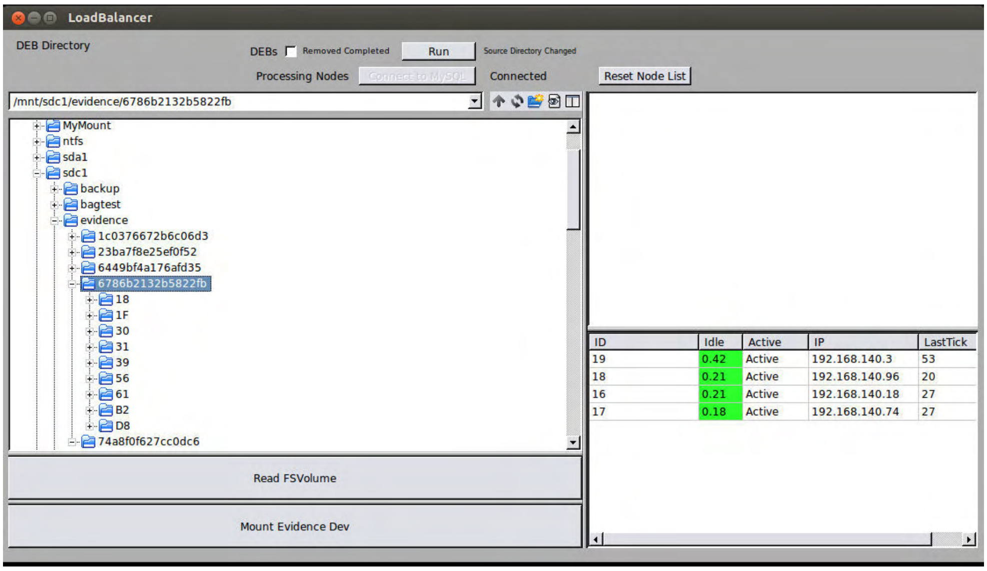
The keytext, in the image File-System Metadata DEC, is searched against the contents of the VolumeListing table; the keytext was generated during the authority stage and should be present in the database table. If it is present, and the record has not expired or has not been previously fulfilled, the import can proceed. If the keytext cannot be found in the VolumeListing table or has expired, it is not possible to import the file-system directory metadata.

The remaining contents of the meta-data DEC is read and the directory meta-data is decrypted and records for each file are created in the inodes, tree and metadata tables. These include fields that describe the full path and filename, file size, MAC dates and times. These fields are marked in the “updated at ingestion stage 1” column in Table 26 - inode table field updates by processing stage.

At the end of this process, a complete ‘framework’ of the directory structure and filenames will have been created in the FClusterfs database. It is actually possible to mount this FClusterfs structure and traverse the directory but as the import of file DECs that contain file data has not yet been carried out there is no actual data to analyse in the files.

A series of “checklists” is now used to control the import of the details and contents of the data-DECs. Table 26 - inode table field updates by processing stage, on page 208, shows the fields set against the stages of ingestion.

Figure 69 shows the initial dialog for data-DEC ingestion.



```
/mnt/sdc1/evidence/6786b2132b5822fb
+ MyMount
+ ntfs
+ sda1
- sdc1
   + backup
   + bagtest
   - evidence
      + 1c0376672b6c06d3
      + 23ba7f8e25ef0f52
      + 6449bf4a176afd35
      - 6786b2132b5822fb
         + 18
         + 1F
         + 30
         + 31
         + 39
         + 56
         + 61
         + B2
         + D8
      - 74a8f0f627cc0dc6
```

ID	Idle	Active	IP	LastTick
19	0.42	Active	192.168.140.3	53
18	0.21	Active	192.168.140.96	20
16	0.21	Active	192.168.140.18	27
17	0.18	Active	192.168.140.74	27

Figure 69 - Stage 1 of Evidence Ingestion

To initiate ingestion, the administrator selects the folder DECs to be used as a staging area for ingestion. This might be mounted removable media or a network connected shared drive into which the acquisition process has deposited the digital evidence containers. The DEC staging area is scanned, any DECs that form part of a Volume that is expected to be imported are found, and the header is read to extract details of the VolumeID, path, filename and size. The inodes table is searched to see if this DEC is expected, that is, there is an entry previously made by a file-system directory metadata DEC import. At this stage, various fields in the inodes table that were created in the Acquisition stage, like the original file's cryptographic hash and staging directory URL, should be empty. If there is a record in the inodes table that satisfies these criteria, then the fields in the table are populated with the meta-data extracted from each of the data-DECs. For each data-DEC, if there is a record in the inodes table and it shows it has already been imported, it will not be considered again.

6.3.4.3 Distribution Assurance

The Distribution section of the Zones schematic is shown in Figure 70

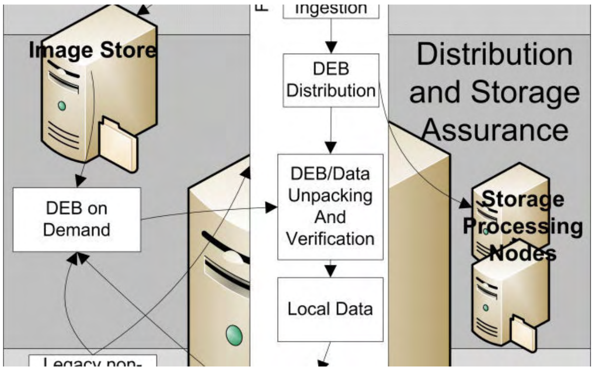


Figure 70 - Distribution Assurance Zone

This stage has four components.

- Load Balancing;
- Moving DECs to their primary destination;
- replication to two other locations;
- unpacking.

6.3.4.3.1 Load balancing

Load Balancing is a function of the ingestion control dialog shown in Figure 69

Having ingested the volume directory metadata the system is now primed to expect the data-DECs that makeup the acquired file system. The first task of load balancing is the selection of the primary storage for the data. It allocates a storage server to hold the data held within the data-DEC and records this in the FClusterfs inodes table. Allocation is based on the available capacity of the host, its processing power and its estimated time to finish its current task list. This data is generated by the cluster-runtime-nodestate daemon and its data is stored in the nodestate table, see Table 24 - nodestate table structure, in section 6.3.4.4.2 and 6.7.3.3.9 for more details on its operation.

6.3.4.3.2 The cluster-runtime-movedata daemon

The cluster-runtime-movedata daemon also uses “checklist” type assurance by constantly scanning the inodes table of FClusterfs for any DEC that has been allocated a primary storage server, not been marked as being ‘in place’ and where the evidence DEC is staged in a directory local to the cluster-runtime-movedata process. This is column three of Table 26 - inode table field updates by processing stage. If these conditions are met, the DEC is transferred to the storage datanode as was allocated by the loadbalancer; see Figure 71. If and only if, the transfer is successful does cluster-runtime-movedata update the inode table with ‘primarystorageinplace’ set to true. cluster-runtime-movedata and cluster-runtime-replicate are the only mechanisms whereby actual data can be moved around the system. It can only operate when all the preconditions from Ingestion Assurance are met. It does not simply scan an evidence folder and move whatever DECs are present; it moves only expected DECs, as recorded in the FCluster inodes table, from a specific folder.

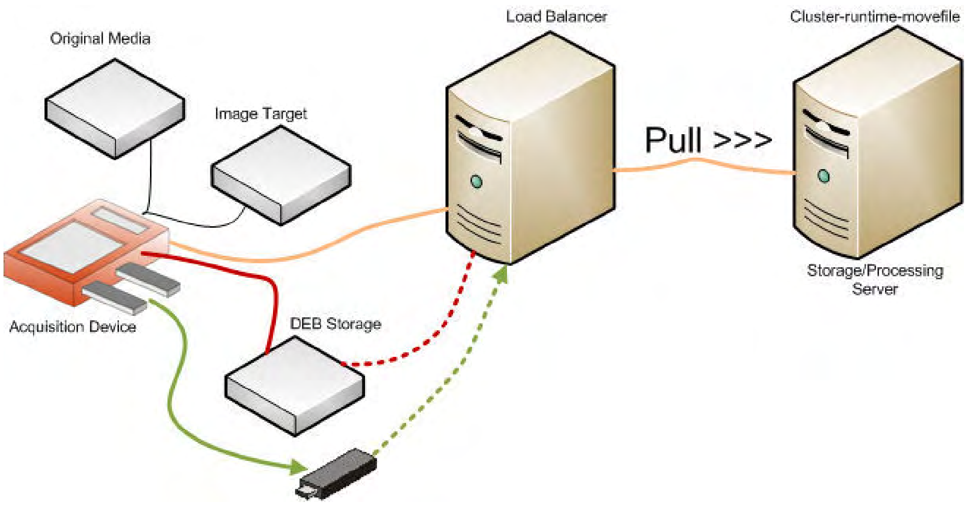


Figure 71 – Movedata

6.3.4.3.3 Cluster-runtime-movefiles Flowchart

The cluster-runtime-movefile daemon controls the placement of DECs in their primary storage position. Data is drawn into the primary server, not pushed by an external agent, based upon data present in the inodes table; see section 6.7.3.3.3 on page 201. The process can be seen in Figure 72 - Cluster-runtime-movefiles Daemon Flowchart.

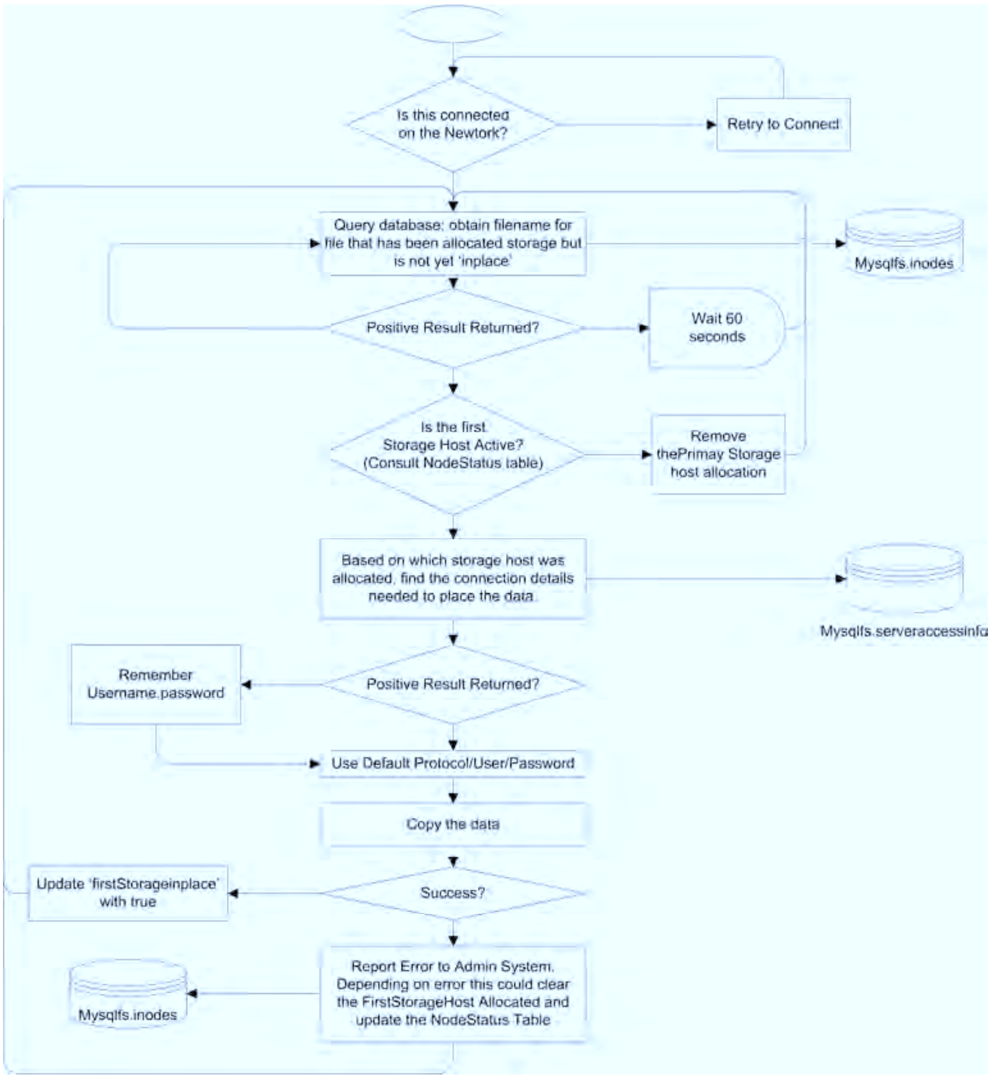

Is this connected on the Newtork?
Retry to Connect
Query database: obtain filename for file that has been allocated storage but is not yet 'inplace'
Mysqlfs.inodes
Positive Result Returned?
Wait 60 seconds
Is the first Storage Host Active? (Consult NodeStatus table)
Remove thePrimay Storage host allocation
Based on which storage host was allocated, find the connection details needed to place the data
Mysqlfs.serveraccessinfo
Remember Username.password
Positive Result Returned?
Use Default Protocol/User/Password
Copy the data
Update 'firstStorageinplace' with true
Success?
Report Error to Admin System. Depending on error this could clear the FirstStorageHost Allocated and update the NodeStatus Table
Mysqlfs.inodes

Figure 72 - Cluster-runtime-movefiles Daemon Flowchart

6.3.4.3.4 The Cluster-runtime-replication daemon

The cluster-runtime-replicate daemon controls the placement of DECs in their secondary and tertiary storage positions. Data is drawn into the storage server from a copy on another storage server, most likely the primary server. It is not pushed by an external agent but pulled by the daemon running within the storage host. This is based upon data present in the inodes table; see section 6.7.3.3.3 on page 201.

6.3.4.3.5 Cluster-runtime-replicate Flowchart

The process can be seen in Figure 73.

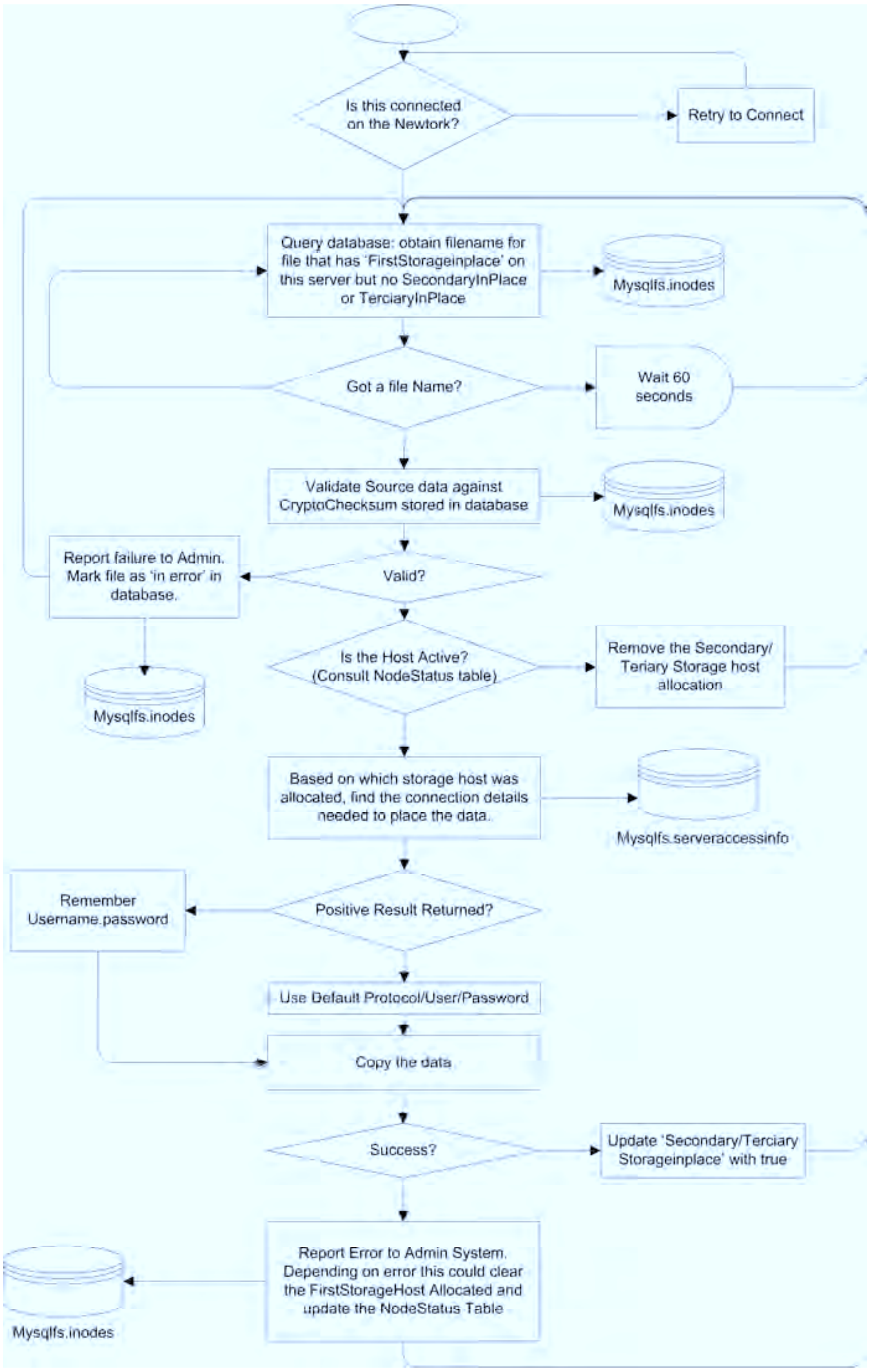

Is this connected on the Newtork?
Retry to Connect
Query database: obtain filename for file that has 'FirstStorageinplace' on this server but no SecondaryInPlace or TerciaryInPlace
Mysqlfs.inodes
Got a file Name?
Wait 60 seconds
Validate Source data against CryptoChecksum stored in database
Mysqlfs.inodes
Report failure to Admin. Mark file as 'in error' in database.
Valid?
Is the Host Active? (Consult NodeStatus table)
Remove the Secondary/ Teriary Storage host allocation
Mysqlfs.inodes
Based on which storage host was allocated, find the connection details needed to place the data.
Mysqlfs.serveraccessinfo
Remember Username.password
Positive Result Returned?
Use Default Protocol/User/Password
Copy the data
Success?
Update 'Secondary/Terciary Storageinplace' with true
Report Error to Admin System. Depending on error this could clear the FirstStorageHost Allocated and update the NodeStatus Table
Mysqlfs.inodes

Figure 73 - Cluster-runtime-replicate Daemon Flowchart

6.3.4.3.6 The cluster-runtime-unpack daemon

Unpacker daemon constantly scans the inodes table to see if there are any DECs that are on its local server but have not been unpacked. It takes the entry from the database and looks to see if the files are on its ftp host, as should be the case from the entries in inodes, not the other way round. A file that simply arrives on the server without an entry in inodes would be ignored. When a suitable DEC is identified, it is split into header and data sections. If this is the primary storage server, the header, containing the meta data read and the meta-data is inserted into the 'meta_data' table and the header file erased. The data section is uudecoded and the data decrypted with a key stored in the VolumeListing table. This was the key first created during Imaging Authority and issued by FCluster to be and used to encrypt the data in the DEC at acquisition time. If the key does not work, the file cannot be decrypted and so unpacking would fail. Only if the file decrypts and the resulting file have a SHA1 checksum that matches both the name of the file itself and the SHA1 as recorded in the inodes table is the data-DEC finally accepted.

6.3.4.3.7 Cluster-runtime-unpackfiles Flowchart

The cluster-runtime-unpack daemon controls the unpacking of DECs and their placement as AIPs in their allotted storage positions. Data is drawn into the storage space by the daemon, not pushed by an external agent, based upon data present in the inodes table; see section 6.7.3.3.3 on page 201. The process can be seen in Figure 74 - Cluster-runtime-unpackfiles daemon Flowchart.

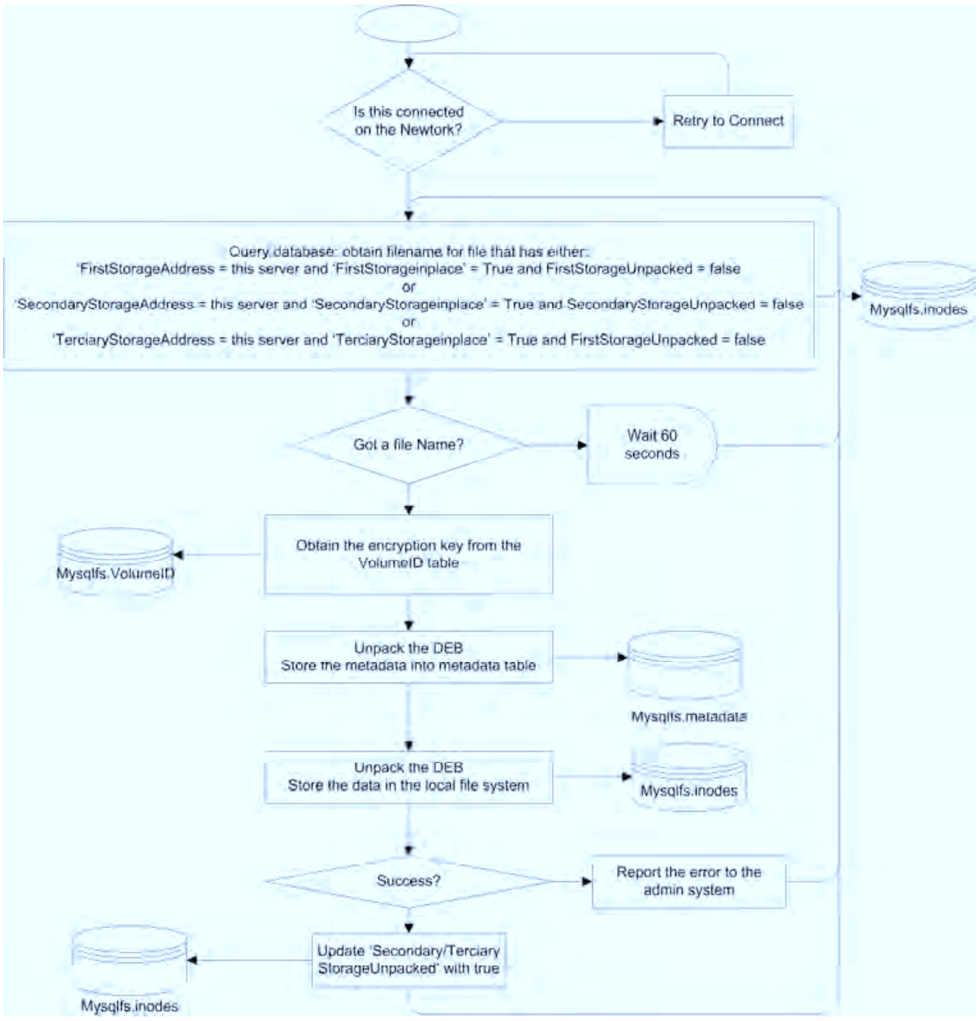

Is this connected on the Newtork?
Retry to Connect
Query database: obtain filename for file that has either:
'FirstStorageAddress = this server and 'FirstStorageinplace' = True and FirstStorageUnpacked = false
or
'SecondaryStorageAddress = this server and 'SecondaryStorageinplace' = True and SecondaryStorageUnpacked = false
or
'TerciaryStorageAddress = this server and 'TerciaryStorageinplace' = True and FirstStorageUnpacked = false
Mysqlfs.inodes
Got a file Name?
Wait 60 seconds
Obtain the encryption key from the VolumeID table
Mysqlfs.VolumeID
Unpack the DEB
Store the metadata into metadata table
Mysqlfs.metadata
Unpack the DEB
Store the data in the local file system
Mysqlfs.inodes
Success?
Report the error to the admin system
Update 'Secondary/Terciary StorageUnpacked' with true
Mysqlfs.inodes

Figure 74 - Cluster-runtime-unpackfiles daemon Flowchart

6.3.4.4 *Processing Assurance*

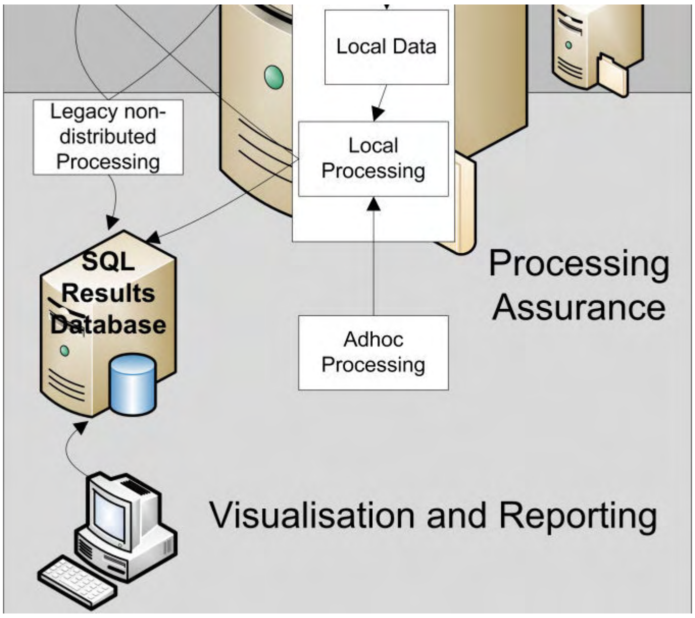


Figure 75 - Processing Assurance Zone

The processing assurance zone, in Figure 75, is the last of the four zones.

The cluster-runtime-processfiles daemon scans the workflow table to see if any jobs are scheduled for a file that is held locally and that it is the designated primary storage for that data. The program is run and the task exit code is stored in the meta-data table.

In section 4.3.5, we categorised programs and assessed the implications of trying to run them in a distributed environment.

It is not possible to cater for programs written in a GUI environment where the code is inherent in the display control and maintain the distributed processing paradigm but FCluster does support this non-distributed processing paradigm, though it is much less effective.

In most cases, the investigator would not see FClusterfs mounted in their user space. Distributed processing is intended to take place on the storage server with local data but this does not prevent FClusterfs being mounted on the user's workstation.

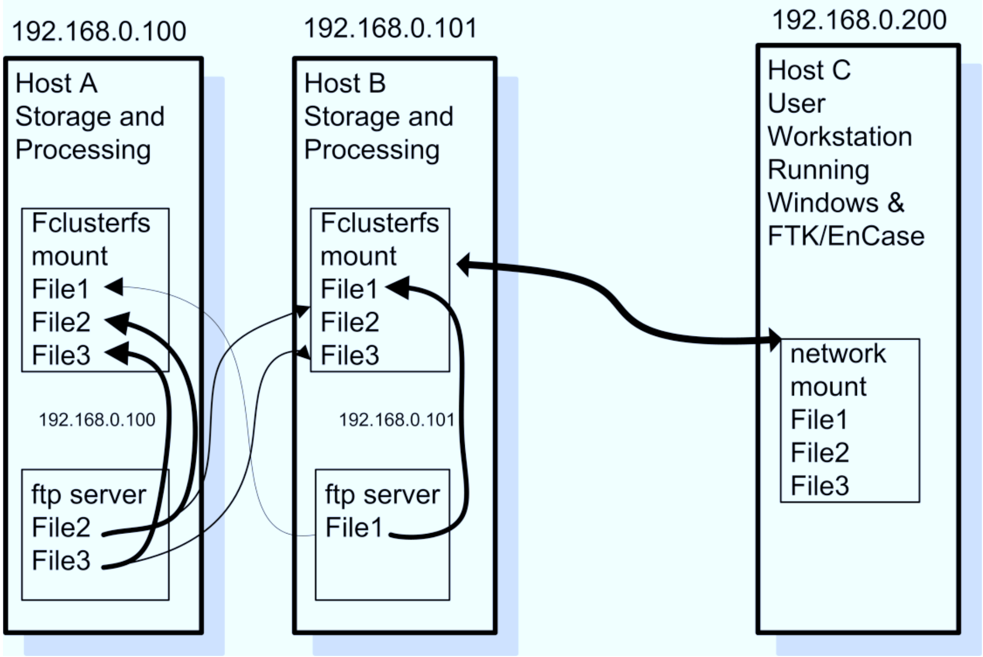


Figure 76 - Legacy GUI Software Access

In this arrangement, shown in Figure 76, any legacy software such as FTK and EnCase can work but will gain no advantage from distributed processing. “File1” could be a legacy image file. All data would have to traverse the network between the storage server and the workstation where the processing will be performed. This is no worse than the centralised network storage model presented in Figure 31 - The Evidence Server and Figure 32 - Case and Evidence Storage both of which are taken from AccessData’s documentation with FTK.

We are assured that the evidence data is correct because all file access must take place by utilising the FClusterfs file-system.

This means that its progress from authority to image, imaging, ingestion, load balancing, primary movement, secondary and tertiary movement, unpacking and constant validation against a cryptographic hash mean that it must be the correct data.

FCluster does not validate the appropriateness or functionality of any application programs that are run but because all programs are normally run from the worktask table it is possible to limit what is acceptable to be placed in the table to a certified subset of commands.

We could, if we wished, control which users can process specific data with specific programs.

FClusterfs also gives us fine grained access control to the files within a file system. by implementing a ‘UserAccessControl’ table FClusterfs could allow and deny access to any file recorded within the database thus allowing some degree of acknowledgement of the rights of privacy of the suspect.

6.3.4.4.1 Cluster-runtime-processfiles Flowchart

The cluster-runtime-processfiles daemon controls the processing of data according to the entries in the workflowtasks table of FClusterfs. This is initiated and controlled by data stored in the inodes table; see section 6.7.3.3.3 on page 201 and the workflowtasks table; see section 6.7.3.3.7 on page 204. The process can be seen in Figure 77 - Cluster-runtime-processfiles Flowchart.

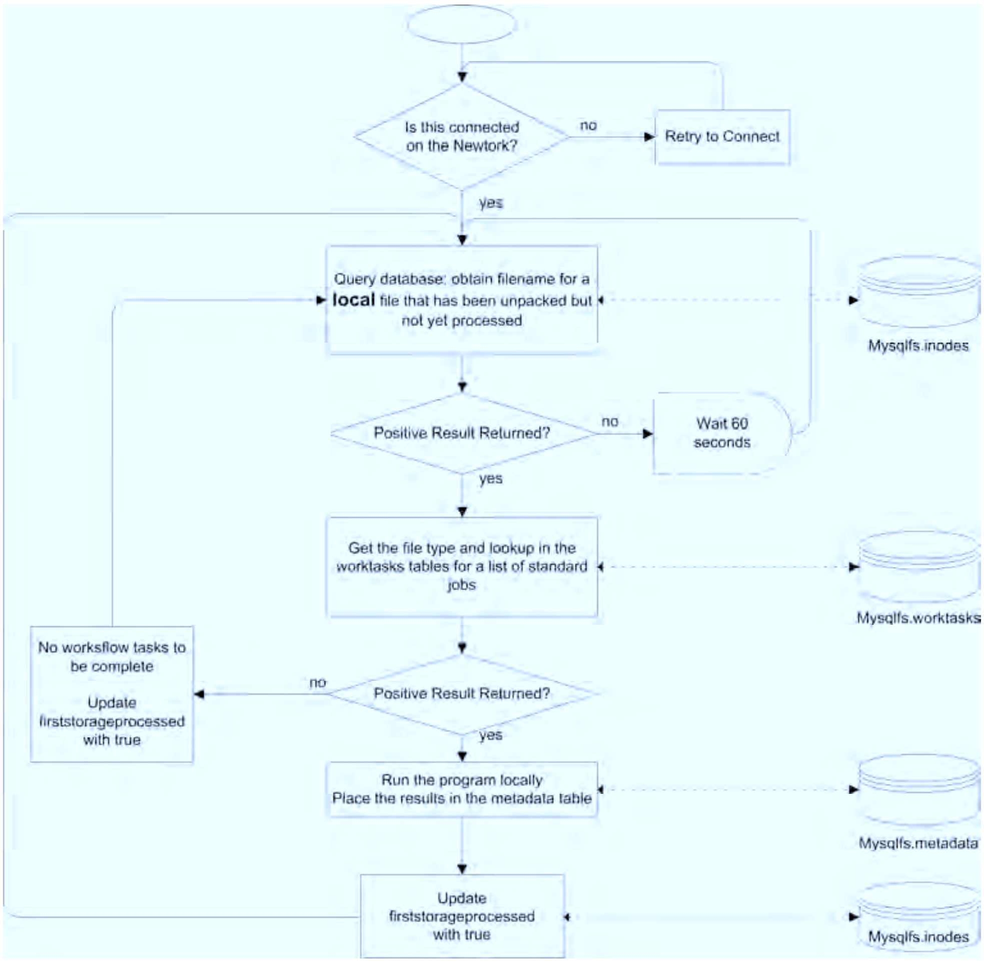

Is this connected on the Newtork?
no
Retry to Connect
yes
Query database: obtain filename for a local file that has been unpacked but not yet processed
Mysqlfs.inodes
Positive Result Returned?
no
Wait 60 seconds
yes
Get the file type and lookup in the worktasks tables for a list of standard jobs
Mysqlfs.worktasks
No worksflow tasks to be complete
Update firststorageprocessed with true
no
Positive Result Returned?
yes
Run the program locally
Place the results in the metadata table
Mysqlfs.metadata
Update firststorageprocessed with true
Mysqlfs.inodes

Figure 77 - Cluster-runtime-processfiles Flowchart

6.3.4.4.2 Cluster-runtime-statemonitor Flowchart

The cluster-runtime-statemonitor daemon controls the processing of local calibration and workload data monitoring and posts the benchmark figures to nodestate table of FClusterfs, see section 6.7.3.3.9 on page 205. The process can be seen in Figure 78 - Cluster-runtime-statemonitor Flowchart.

An instance of the cluster-runtime-nodestate daemon program, written in Bash, runs on each Cluster member. During a run, it gathers information about key features of the host. cluster-runtime-statemonitor calls a variety of utility programs to gather operational statistics about the host computer. These include processing load on each core, available and committed

RAM, available and spare data storage, the network link speed. This data is sent to the NodeState table that contains records of each of the nodes within the cluster. The Load Balancing process refers to the data in the NodeState table to determine which storage server will be the primary holder of the data.

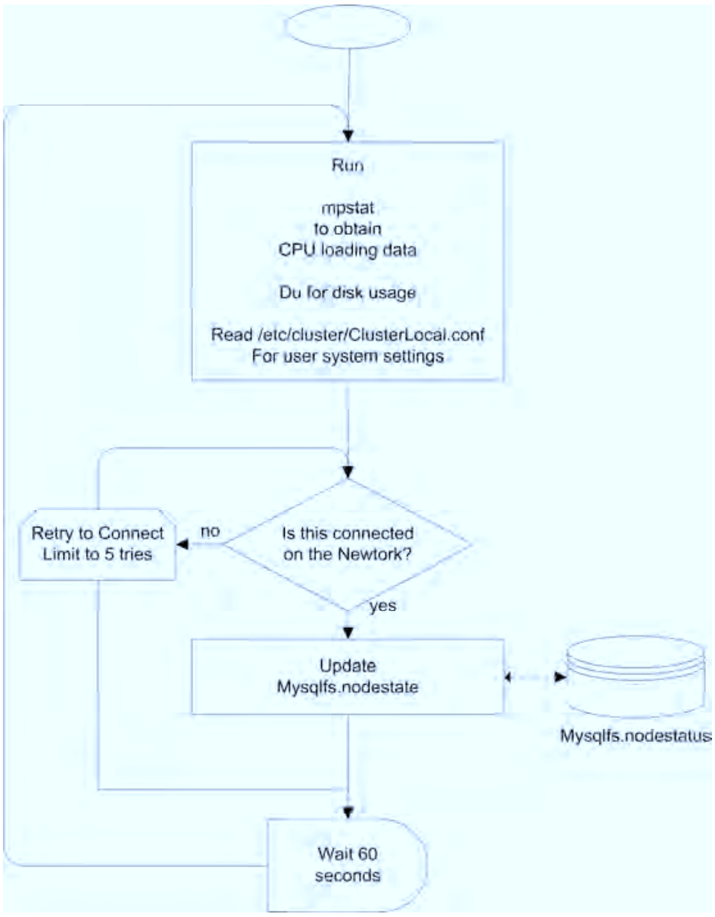


Figure 78 - Cluster-runtime-statemonitor Flowchart

At a set time interval, perhaps every 24 hours, and certainly when any key feature of the system is changed, each node is 'calibrated' by running a specific set of test programs with a standardised 'typical' data set. The time taken to complete each of these calibration tasks is stored locally and is sent to the MySQL server and recorded in the NodeState table, 6.7.3.3.9, Table 24 - nodestate table structure.

The Estimated Time to Complete (ETC) is calculated by factoring the processing time per megabyte for a specific file extension against the time last recorded as the time taken to complete a megabyte of processing of that file type. This would yield an indicative ETC, which until the task is undertaken, would be a good a guide as any.

The selection of a storage/processing server to hold the primary data copy is based upon the ETC relative to the ETCs from other hosts. In this way, hosts that are more powerful are subject to a greater utilisation.

6.4 Jigsaw Imaging

6.4.1 Introduction

In the previous section, 6.3, we described FCluster and made a design decision that to attain higher assurance, departing from current practice, the imaging of source media should be an integrated part of the forensic system. In this section we describe "jigsaw Imaging"; a novel approach to forensic imaging which is more suited to very large media and concurrent processing of data. This approach is based on the observation that when executing a raw linear type of imaging on a machine with an i7 processor, the processor is vastly underutilised and could do much more. We reviewed existing approaches to imaging in section 4.5

6.4.2 The Jigsaw Technique

6.4.2.1 Overview

The primary objective in Jigsaw Imaging is to obtain usable data while processing a conventional imaging process. In LOTA – "Bingo Imaging" in 4.6.1.3, Roussev, Quates and Martell proceeded traditional linear imaging by first reading the file system meta-data from the directory and using this data to monitor the completion of files as the data was imaged sector by sector in a linear fashion. In this way, LOTA is able to identify which files have been completed by the linear copy process and make them available for analytical processing. We move further in that we use the file system data to control the sequence in which the sectors are processed.

In our technique, which we have named "Jigsaw Imaging", we develop on existing techniques by tracking the data through paths defined by the location of file data as recorded in the file-system directory rather than simply reading sequentially through the media sector by sector. Overall, the production of an image will take longer but Jigsaw makes investigable evidence available almost immediately.

6.4.2.2 Strategy

Our strategy will be familiar to anyone who has ever attempted to build a jigsaw puzzle. While a small jigsaw of perhaps 20 pieces can be conquered by starting at the top left and locating each piece sequentially by matching from the picture, this approach becomes untenable when applied to puzzles of thousands of pieces. The alternative is obvious. First, we search for all the pieces with a straight edge, in particular we look for the corners. When we have found and positioned the corners, we build a rectangular frame into which we will place all subsequent pieces. Next, we look at the pile of remaining pieces and try to group them into piles of similar colours or patterns. We attempt to complete these areas 'in isolation' and having done so, we complete the remaining areas as best we can fill in the gaps between the areas already completed.

Although not all areas of the hard disk allow us to apply this structured approach, where we can, we use the information structures on the media to direct our imaging process we do but in other areas, we adopt linear imaging. Our strategy works in areas of the disk that contain known file-system formatted data where we are able to track through the data and read data in a sequence that yields analysable data as quickly as possible. It does not work in areas like the partition table, HPA, DCO or unused or at least unidentified space on the media. In these, we revert to a linear sequence.

The prototype program only addresses devices with the NTFS file-systems and so we will be specific in this. We will not explain the intricacies of the NTFS file-system. We refer the reader to File System Forensic Analysis (MySQL 2014) and ntfs.com (MySQL 2014) for more details of NTFS but will highlight some key features.

System File	File Name	MFT Record	Purpose of the File
Master file table	$Mft	0	Contains one base file record for each file and folder on an NTFS volume. If the allocation information for a file or folder is too large to fit within a single record, other file records are allocated as well.
Master file table 2	$MftMirr	1	A duplicate image of the first four records of the MFT.
Log file	$LogFile	2	Contains a list of transaction steps used for NTFS recoverability. Log file size depends on the volume size and can be as large as 4 MB. It is used by Windows NT/2000 to restore consistency to NTFS after a system failure.
Volume	$Volume	3	Contains information about the volume, such as the volume label and the volume version.
Attribute definitions	$AttrDef	4	A table of attribute names, numbers, and descriptions.
Root file name index	$	5	The root folder.
Cluster bitmap	$Bitmap	6	A representation of the volume showing which clusters are in use.
Boot sector	$Boot	7	Includes the BPB used to mount the volume and additional bootstrap loader code used if the volume is bootable.

System File	File Name	MFT Record	Purpose of the File
Bad cluster file	$BadClus	8	Contains bad clusters for the volume.
Security file	$Secure	9	Contains unique security descriptors for all files within a volume.
Upcase table	$Upcase	10	Converts lowercase characters to matching Unicode uppercase characters.
NTFS extension file	$Extend	11	Used for various optional extensions such as quotas, reparse point data, and object identifiers.
		12-15	Reserved for future use.
Quota management file	$Quota	24	Contains user assigned quota limits on the volume space.
Object Id file	$ObjId	25	Contains file object IDs.
Reparse point file	$Reparse	26	This file contains information about files and folders on the volume include reparse point data.

Figure 79 - NTFS Metadata Files (NTFS.com)

In an NTFS file system all data is contained in files, even the directory of the file system itself. There are about 25 special files that contain file system meta-data. These files are always the first written when the file system is created and occupy the first 32 ‘slots’ in the directory table. These files all begin with the “$” symbol and are all hidden to ‘normal’ computer users. Unlike other files, the system files entries are always in a defined position and never move. See Figure 79 - NTFS Metadata Files (NTFS.com)

The most important file-system meta-data is contained within a system file called $MFT which is always $MFT record 0. Another note-worthy file is $Bitmap, which is always MFT record 6, which contains a bitwise representation of which cluster are in use and which are currently unallocated to files. Each of the records in $MFT contains a data item which points to the cluster at which the co-responding data starts and any defragmentation which occurs.

6.4.2.3 Jigsaw Imaging Deliverables

Jigsaw imaging has two types of deliverables – (Figure 80 - Jigsaw Device Connection).

- An imaged copy or copies of the original;

- A collection of Digital Evidence Containers that contain the potential evidential data. There are two types of DEC contents.
  - A single master DEC containing specifically directory meta-data describing the contents of the second type of DEC, shown in Figure 88 - FCluster Master DEC;
  - A Multitude of DECs containing the actual data, shown in Figure 89 - FCluster Data DEC.

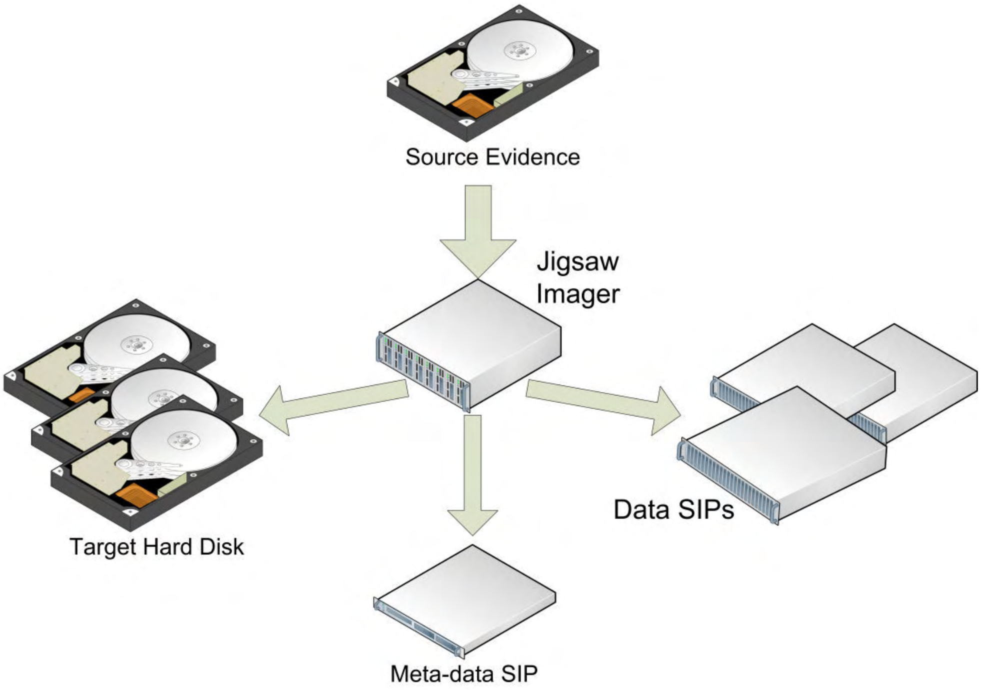


Figure 80 - Jigsaw Device Connection

There could be a multitude of files forming one data DEC or a multitude of data DECs each containing one file or a combination of multiple files in multiple data DECs, or a combination of the pervious arrangements.

In our description, we use the generic term DEC. This can be any one of a variety of formats suitable for storing one of more files. AFF is an obvious choice.

6.4.2.4 Outline

Jigsaw imaging uses linear imaging but directs this by interpreting the file-system and using this information to direct the focus to the high value areas of the media; namely those with actual data. Jigsaw imaging first accesses the disk meta-data, then the data within each partition and then, in turn, the directory data for each partition. Referring back to the description of our strategy in 6.4.2.2, this is like building the *borders* of the file-system into which we can fill the gaps with data from the files. Then we access the files themselves.

Once we have read the files, we can then read the unallocated space. See Figure 81 - Jigsaw Imaging Process

As we have read the directory, we can apply a selection criterion to select certain files to be accessed as a priority because we consider them likely to hold evidence, see 6.4.2.5. Jigsaw imaging reads these files and adds them to the evidence DEC(s). It then reads all the other files, considered to be of lesser potential value by or selection criterion, and finally reads the unallocated space to complete the imaging of the partition.

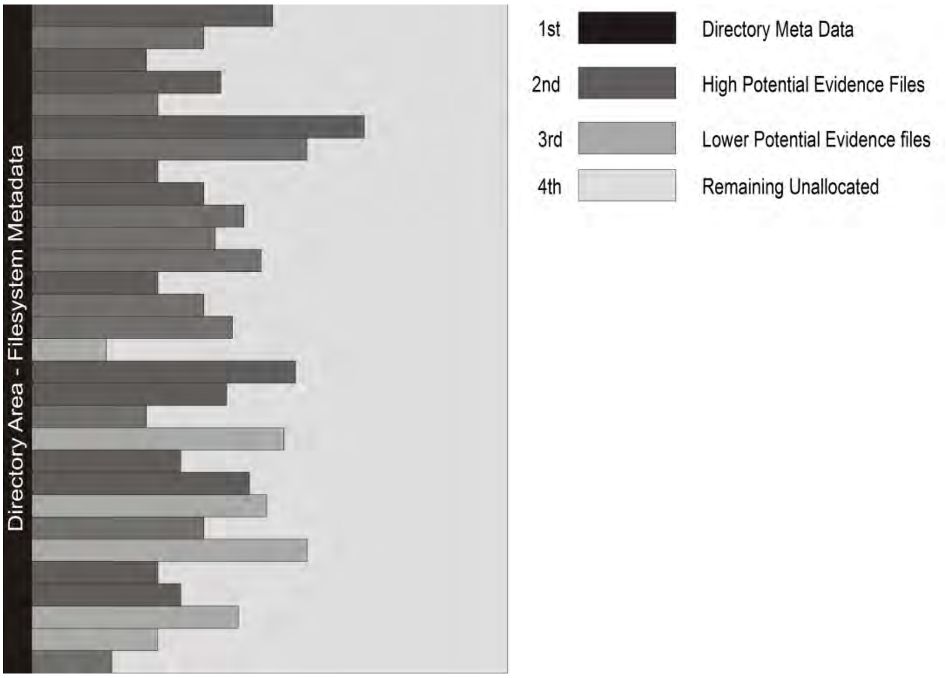

Directory Area - Filesystem Metadata
1st
Directory Meta Data
2nd
High Potential Evidence Files
3rd
Lower Potential Evidence files
4th
Remaining Unallocated

Figure 81 - Jigsaw Imaging Process

The most important single feature of Jigsaw imaging is that as each block/cluster is read from the source, it is written in its corresponding location to the image target drive(s) while simultaneously creating DECs of the file. Each block/cluster is read just once from the source and written just once to the target drive. This is described in Figure 82 - Jigsaw Read Write Sequence.

The nature of this process means that the image output cannot be to a file on a file system as is the case with EWF and FTK, but is written as an image, written directly to a storage device. As each cluster is read from the original, it must be written directly to the corresponding cluster on the image target.

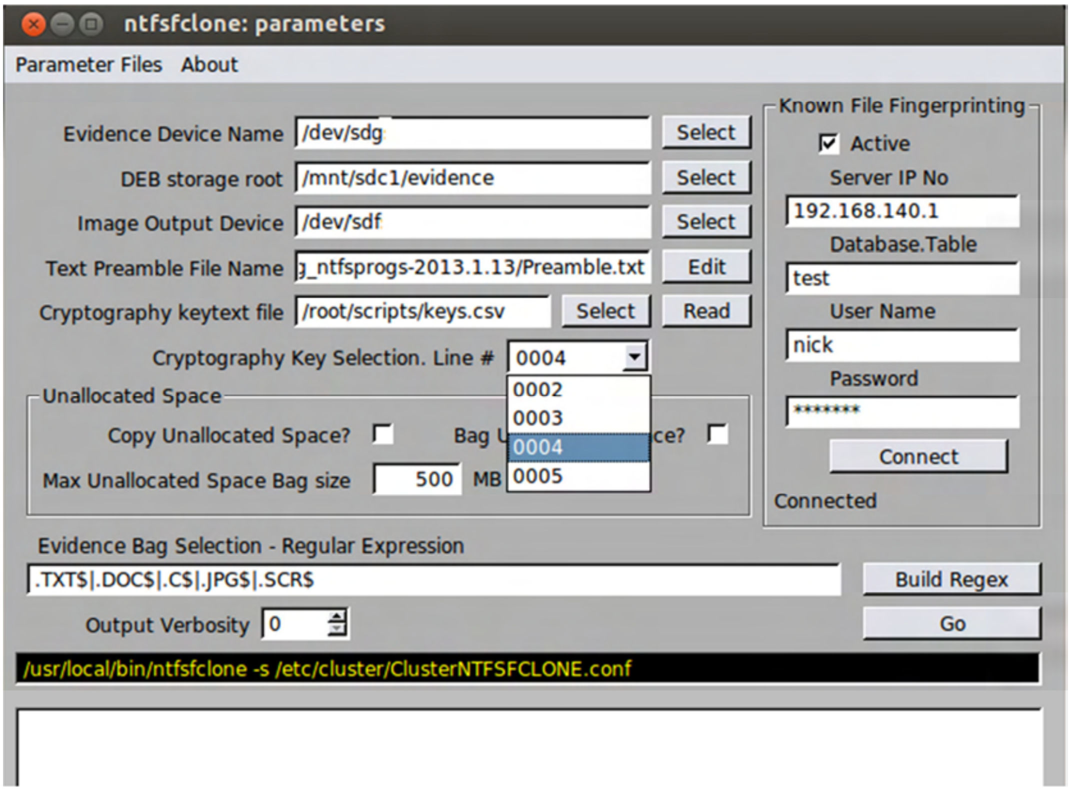
```
Select a Partition
Select a file
Create a new DEC

Read cluster from Source media
Write to corresponding location on target media
Append to DEC(s)
Further Writes as required

Read cluster from Source media
Write to corresponding location on target media
Append to DEC(s)
Further Writes as required

Read cluster from Source media
Write to corresponding location on target media
Append to DEC(s)
Further Writes as required

Close DEC
```

Figure 82 - Jigsaw Read Write Sequence

6.4.2.5 Selecting data; Primary and Secondary Evidence

The prototype includes a simple filtering facility where a regular expression can be given to define a subset of files to be 'bagged' as DECs. In Figure 83 - Acquisition initiation Dialog, we are choosing as files ending in ".txt", ".doc", ".c", ".jpg" and ".scr" as an example. The prioritisation technique, described in section 6.5, on page 181, greatly enhances this.

6.4.2.6 *Initiation*



Evidence Device Name: /dev/sdg

DEB storage root: /mnt/sdc1/evidence

Image Output Device: /dev/sdf

Text Preamble File Name: g_ntfsprogs-2013.1.13/Preamble.txt

Cryptography keytext file: /root/scripts/keys.csv

Cryptography Key Selection. Line #: 0004 (0002, 0003, 0004, 0005)

Copy Unallocated Space? □

Bag U[...]ce? □

Max Unallocated Space Bag size: 500 MB

☑ Active

Server IP No: 192.168.140.1

Database.Table: test

User Name: nick

Password: *******

Connected

Evidence Bag Selection - Regular Expression: .TXT$|.DOC$|.C$|.JPG$|.SCR$

Output Verbosity: 0

```
/usr/local/bin/ntfsfclone -s /etc/cluster/ClusterNTFSFCLONE.conf
```

Figure 83 - Acquisition initiation Dialog

The operation of Jigsaw Imaging is best explained with the features of the initiation dialog box, Figure 83 - Acquisition initiation Dialog. The options are explained in Figure 84 - Jigsaw Imaging Options

Evidence Device Name	the name of the source media. This is the name of a storage device, in this case /dev/sdg (not a partition on that drive which would be /dev/sdg1)
DEC Storage root	The root directory, in a file system, where the DECs will be deposited as they are created.
Destination drive for the image	The destination name of the media. As with the Evidence Device Name, this is the name of a storage device, in this case /dev/sdg (not a partition on that drive which would be /dev/sdg1)
Text preamble file name	This should contain the name of a text file whose content describes meta-data for this imaging process. This enables a text note to accompany the image meta-data.
Cryptographic Keytext file	See section 6.4.2.7
Cryptographic Key Selection	See section 6.4.2.7
Unallocated Space Copy Unallocated Space?	Will the copy continue into the unallocated space of the file system?
Unallocated Space Bag Unallocated Space?	If the previous option is selected, will the space be bagged into DECs?
Unallocated Space Max Unallocated Space Bag Size	If the previous option is selected, what size will the bags be?
Known File Fingerprinting	This is the connection information of an SQL server that can be used as a reference to test file fingerprints to enable 'known safe' or 'known unsafe' to be ignored or highlighted as appropriate. This can be enabled or disabled as required.
Evidence Bag Selection	This can contain a regular expression used to define which subset of files to be bagged as the imaging process proceeds.

Figure 84 - Jigsaw Imaging Options

6.4.2.7 Jigsaw as part of the FCluster system

The motivation that led to Jigsaw imaging was twofold. Firstly, we feel that FCluster needed a corresponding imaging technique that aspired to a higher-level assurance found in FCluster and secondly that we needed an imaging technique that lent itself to yielding the advantages so long promised by parallel distributed processing but not delivered.

Unlike any existing system, in FCluster imaging starts within FCluster itself. In the first of the four assurance zones that we first introduced in Figure 55**Error! Reference source not found.**, an authority to image is generated by the action of an administrator when they trigger the creation of a set of cryptographic hashes destined to be sent to, and used by, a specific imaging device, shown in Figure 64- Acquisition Authority Dialog Box. This process is detailed in section 6.3.4.

6.4.2.8 Exit Points

Jigsaw imaging is intended to provide the investigator access to potential evidence as soon as possible whilst still completing a traditional image in an acceptable time. This is facilitated by having 'exit points' within the full sequence.

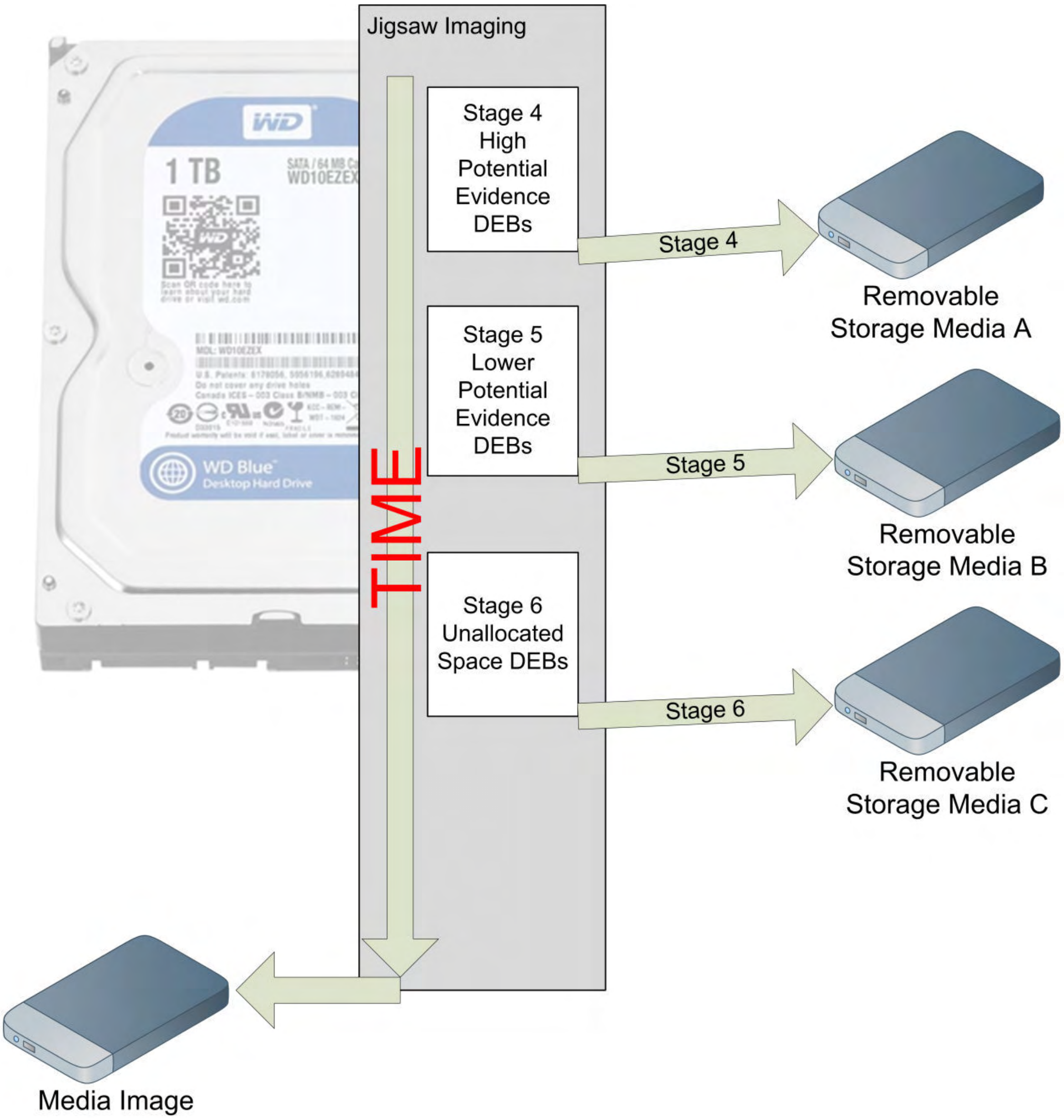

Jigsaw Imaging
Stage 4 High Potential Evidence DEBs
Stage 5 Lower Potential Evidence DEBs
Stage 6 Unallocated Space DEBs
TIME
Stage 4
Stage 5
Stage 6
Removable Storage Media A
Removable Storage Media B
Removable Storage Media C
Media Image
1 TB
WD Blue
Desktop Hard Drive

Figure 85 - Jigsaw Exit Points

Figure 85 - Jigsaw Exit Points shows that after the high potential evidence has been captured in stage 4, the 'removable storage media A' can be removed and passed to the analysis software. Meanwhile, Jigsaw imaging will continue until stage 5 is complete, at which point the 'removable media B' can be removed and sent for analysis. If required, Stage 6 can write the unallocated space as DECs to 'removable media C'. It may be some time after the three pervious captures have completed that the image hard disk is available.

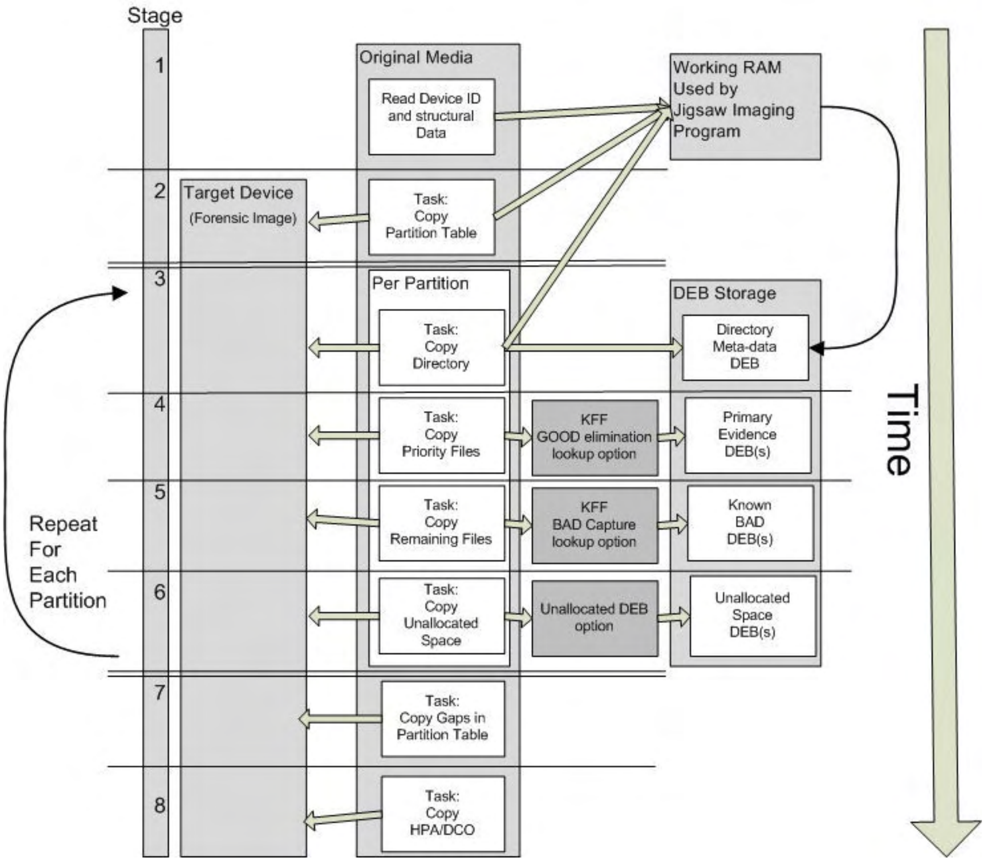

Stage
1
2
3
4
5
6
7
8
Original Media
Read Device ID and structural Data
Working RAM Used by Jigsaw Imaging Program
Target Device (Forensic Image)
Task: Copy Partition Table
Per Partition
DEB Storage
Task: Copy Directory
Directory Meta-data DEB
Task: Copy Priority Files
KFF GOOD elimination lookup option
Primary Evidence DEB(s)
Task: Copy Remaining Files
KFF BAD Capture lookup option
Known BAD DEB(s)
Repeat For Each Partition
Task: Copy Unallocated Space
Unallocated DEB option
Unallocated Space DEB(s)
Task: Copy Gaps in Partition Table
Task: Copy HPA/DCO
Time

Figure 86 - Jigsaw Process Flow

6.4.2.9 *The Sequence in detail*

There are eight stages in Jigsaw Imaging. Shown in detail in Figure 86 - Jigsaw Process Flow.

- Stage 1 reads data from the device control channel to identify the storage media as a device;
- Stage 2 uses linear imaging to read the partition table; actually, this is just one sector;
- Stages 3 to 6 use file-directed linear imaging to read the contents of a recognised file system;
- Stages 7 and 8 use linear imaging to read the remainder of the storage device.

6.4.2.9.1 Stage 1 - The Storage Device

First, the device ID and structural data, about LBAs, is read from the source device. This is not actually part of the media contents and so is not written to the target device but will be written, as part of the header section, to a “directory metadata DEC” file.

6.4.2.9.2 Stage 2 - MBR, the partition table

The Master Boot Record, which contains the partition table, is read and interpreted as a guide for the rest of the imaging process. All currently agreed standards for MBR are set at a single block of 512 bytes. As the partition information is read, it is written to the corresponding block on the image target drive and is also interpreted and stored in RAM. The appended to the directory metadata DEC file as part of the header section.

6.4.2.9.3 Stage 3 – Directory within a partition

When starting to acquire a file-system partition, Jigsaw imaging starts by reading the directory, which in the case of an NTFS partition is the $MFT file, which always starts in cluster 0, and creates an array of data representing the file structure of the partition. Directory entries in NTFS are usually 1024 bytes (Carrier 2005) and so even large disks rarely have $MFT files greater than 500MB. As with the partition table data, as each block/cluster of data is read, it is written to the corresponding cluster on the target drive and is appended to the directory metadata DEC file as part of the header section. The $Volume MFT Record is read to obtain the volume label and version information which is written to the Directory DEC. The $MFT file contains, not only, entries for ‘visible’ files which can be seen by the user but also a set of up to 32 ‘system’ files which hold information about the file-system’s meta-data that includes, for example, security schemas, unused cluster availability. These files are copied in section 6.4.2.9.6

6.4.2.9.4 Stage 4 –Selected/Prioritised Files with a partition

In Jigsaw imaging, the array of file/directory data created in the previous stage is used to select files that conform to a selection criteria decided by the acquirer. In the prototype, these can be selected by a regular expression acting on the file name or a selection based on the MAC dates of the file.

There is an improved File Selection and processing Prioritisation in section 6.5

Jigsaw imaging proceeds by processing each of the selected files. A cluster-by-cluster copy of the file is taken to the target drive at the same time a digital evidence container is built on the DEC storage media. Each cluster that makes up these files is read only once from the source and written at least twice, once to the image target and once to the DEC.

It is most likely that the End Of File will be encountered before the end of the final cluster. Jigsaw Imaging includes the data in slack space in its copy.

As a file is read, Jigsaw imaging calculates a cryptographic checksum for the file. This is possible because, unlike previous imaging techniques, Jigsaw reads by following through files not clusters. In the prototype, there is an option to lookup these checksums against a database of Known File Fingerprints. As the files are read, the SHA1/MD5 is calculated.

Known ‘Good’ files can be discarded at this stage and not included in the primary DEC collection. Also known bad hashes can trigger an alert to the Acquirer for appropriate action to be taken.

6.4.2.9.5 Jigsaw imaging - exit point Stage 4

At the end of this stage, all of the evidence selected as *primary* will have been collected in DECs. In Figure 80 and Figure 85 we showed that several storage devices could be used for DECs. If a device had been allocated to storing the DECs produced by stage 4, it could now be removed and the transport and ingestion to the analysis facility could begin leaving the remainder of the imaging process to continue. More so, if the DECs were stored on shareable media, they could be ingested and processed as soon as they are created.

6.4.2.9.6 Stage 5 - Residual files

After the primary evidence files are copied then the remaining files are imaged. This stage includes copying the remaining NTFS system files that occupy inodes less than 32. As with the previous files, these are read cluster by cluster and written cluster by cluster to the target media. As the clusters pass through, the cryptographic hash is calculated. If selected, the result can be tested against a database of SHA1 checksums of files known to be ‘bad’ and the acquirer can be alerted.

6.4.2.9.7 Jigsaw imaging - exit point Stage 5

At the end of this stage, provided the option was selected, all of the evidence selected as *‘bad’* from its KFF will have been collected in DECs. In a similar way to ‘DEC exit point Stage 4’, this data can be processed as the rest of the imaging continues.

6.4.2.9.8 Stage 6 - Unallocated Space

The NTFS system file, $volume, contains a bit mapping of cluster usage. In this final stage, it is used to copy, sequentially cluster by cluster, all the remaining data from the original to the target.

Jigsaw imaging has an option to write unallocated space data to DEC(s). This is a significant departure from current practice. Current carving software, known to the author, only supports carving of whole image files. Jigsaw imaging can provide a series of DECs that contain the data from unallocated areas of the files system. The advantage of this is that it can be used across a distributed system to allow parallel processing of carving without special programming techniques.

During our experiments for prioritisation, we found a number of significant factors that affect Unallocated Space.

- Most files, 97%, are not fragmented at all, 1 or 2 are very fragmented (500+ fragments);
- We note that Windows XP does not automatically run defrag;
- Windows 7 and above schedules automatic defrag runs for hard disks;
- Windows 7 and above does not automatically defragment Solid State Drives;
- Solid State Drive often clear sectors that held data and have been deleted in the file-system.

File slack space is read within the Files Section See 6.4.2.9.4

6.4.2.9.9 Jigsaw imaging - exit point Stage 6

At this stage, all the unallocated space data will have been written to DECs. In a similar way to ‘DEC exit point Stage 4 and Stage 5’, this data can be processed as the rest of the imaging continues.

Stages 3 – 6 are repeated for each partition found in stage 2 provided it is of a known format so that the Jigsaw imaging program can interpret its structure. If an unknown partition type is discovered then linear imaging is more appropriate.

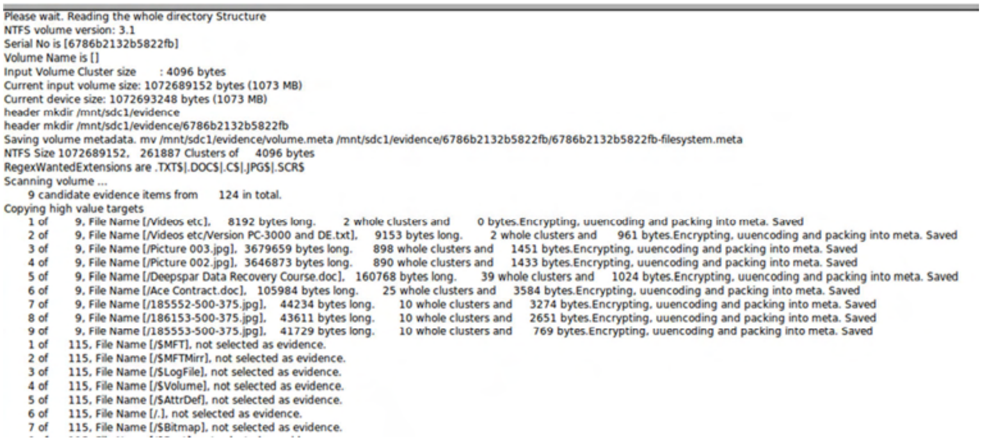
6.4.2.9.10 Copy Gaps in Partition Table

Linear imaging is used to copy any gaps left between the partitions. This will be evident from the partition table data.

6.4.2.9.11 HPA and DCO

Finally, the HPA/DCO areas are copied cluster by cluster to the target using linear imaging.

An audit trail can be seen in Figure 87 - Jigsaw Imaging Progress.



```
Please wait. Reading the whole directory Structure
NTFS volume version: 3.1
Serial No is [6786b2132b5822fb]
Volume Name is []
Input Volume Cluster size      : 4096 bytes
Current input volume size: 1072689152 bytes (1073 MB)
Current device size: 1072693248 bytes (1073 MB)
header mkdir /mnt/sdc1/evidence
header mkdir /mnt/sdc1/evidence/6786b2132b5822fb
Saving volume metadata. mv /mnt/sdc1/evidence/volume.meta /mnt/sdc1/evidence/6786b2132b5822fb/6786b2132b5822fb-filesystem.meta
NTFS Size 1072689152,  261887 Clusters of    4096 bytes
RegexWantedExtensions are .TXT$|.DOC$|.C$|.JPG$|.SCR$
Scanning volume ...
    9 candidate evidence items from     124 in total.
Copying high value targets
    1 of      9, File Name [/Videos etc],    8192 bytes long.       2 whole clusters and       0 bytes.Encrypting, uuencoding and packing into meta. Saved
    2 of      9, File Name [/Videos etc/Version PC-3000 and DE.txt],    9153 bytes long.       2 whole clusters and     961 bytes.Encrypting, uuencoding and packing into meta. Saved
    3 of      9, File Name [/Picture 003.jpg],  3679659 bytes long.     898 whole clusters and    1451 bytes.Encrypting, uuencoding and packing into meta. Saved
    4 of      9, File Name [/Picture 002.jpg],  3646873 bytes long.     890 whole clusters and    1433 bytes.Encrypting, uuencoding and packing into meta. Saved
    5 of      9, File Name [/Deepspar Data Recovery Course.doc],  160768 bytes long.      39 whole clusters and    1024 bytes.Encrypting, uuencoding and packing into meta. Saved
    6 of      9, File Name [/Ace Contract.doc],  105984 bytes long.      25 whole clusters and    3584 bytes.Encrypting, uuencoding and packing into meta. Saved
    7 of      9, File Name [/185552-500-375.jpg],   44234 bytes long.      10 whole clusters and    3274 bytes.Encrypting, uuencoding and packing into meta. Saved
    8 of      9, File Name [/186153-500-375.jpg],   43611 bytes long.      10 whole clusters and    2651 bytes.Encrypting, uuencoding and packing into meta. Saved
    9 of      9, File Name [/185553-500-375.jpg],   41729 bytes long.      10 whole clusters and     769 bytes.Encrypting, uuencoding and packing into meta. Saved
    1 of    115, File Name [/$MFT], not selected as evidence.
    2 of    115, File Name [/$MFTMirr], not selected as evidence.
    3 of    115, File Name [/$LogFile], not selected as evidence.
    4 of    115, File Name [/$Volume], not selected as evidence.
    5 of    115, File Name [/$AttrDef], not selected as evidence.
    6 of    115, File Name [/.], not selected as evidence.
    7 of    115, File Name [/$Bitmap], not selected as evidence.
```

Figure 87 - Jigsaw Imaging Progress

6.4.2.10 Parallel carving using DECs

Simply put, carving involves reading a stream of data and attempting to detect data that can be recovered in some intelligible form. There are several factors that govern the success of carving (Richard III & Roussev 2005) among them are:

- Fragmentation – carving can only recover whole files when they are stored in contiguous clusters and have not been overwritten in part. It can recover part files consisting of a series of contiguous cluster whose run is interrupted when clusters are overwritten;

- Analysis complexity – in general, the more complex the regular express or search technique, the slower the process. For different searches, often multiple passes are needed;

- The read speed of the storage media contributes to whether this is a disk I/O bound, where the disk cannot keep up with the need to feed the processor, or processor bound operation, where the processor leaves the media I/O idling.

Jigsaw imaging can collect data from unallocated space in DECs. The DECs can then be submitted to FCluster for analysis with existing tools such as Scalpel (Richard III & Roussev 2005). It is very unlikely that all the unallocated space within a file system will occupy one large contiguous run of clusters; although this could be the case after a thorough defragmentation of the disk. It is most likely that it will be interspersed by the data from a ‘real’ file. It is far more likely, the unallocated space will comprise of a scattering of runs of clusters of a variety of sizes.

Jigsaw imaging does allow DECs to be built with any number of clusters but in practice, they need to be limited in size to enable them to be processed effectively across a distributed processing system.

In Jigsaw imaging, the unallocated space DECs can be limited in the maximum size to which they are allowed to grow. A natural break is when a ‘run’ of unallocated clusters ends when it is interrupted by a cluster which is marked as ‘in use’; this data is readily available from the NTFS file $Volume. Another possible break point is when the user specifies a maximum file size. The default length set for Jigsaw is an arbitrary 500MB but can be changed. 500MB is based on our research that shows that, typically, 40% of all files on a system running a Windows OS are less than 10MB.

Carving has two objectives; to recover entire files, for example photographic images, or smaller artefacts that are like email addresses or credit card numbers.

In a 500MB file with 4k clusters, there are 125,000 possible start positions for a file because they must be cluster aligned. If a high resolution JPG is 10MB it will occupy 2,500 clusters. This means that a randomly placed 10MB will be truncated only if it starts in any one of the last 2500 clusters. A chance of 2,500/125,000 or 1:50. A 1GB DEC would reduce this chance to 2,500/250,000 = 1:100, and so on. An email address might be as many as 50 characters but could start anywhere within a cluster. In this case it could be partially lost if it started in the last 50 characters of a DEC. A chance of 50 in 500,000,000. The latter is so small as to be of negligible risk, if the former is considered too great then the acquirer should choose a larger DEC size when using Jigsaw imaging.

6.5 A simplified SIP, DEC design

We discussed the current availability of Digital Evidence Container designs in section 4.6.1, on page 118. With the assumption of the unsuitability of image formats like those create by dd to cope with ever larger media simply because of size, we need to make some design decisions about a DEC upon which a distributed architecture can be based.

It is clear that because of its provision for reference to external data and its potential AFF4 is the most likely candidate upon which an implementation of FCluster should be based but we consider AFFx too complicated to implement in a prototype. The metadata that forms the header part of the simplified DEC is very similar to the output of fiwalk command found in the Sleuthkit (Carrier 2015) but the Sleuthkit only works on forensic images of media not individual files themselves.

Instead, for development purposes only, we substitute a very basic design that enables us to demonstrate the action of assurance in FCluster, yet it embodies all the principles needed in a DEC.

In FCluster, data is packaged in two types of DECs; a Master DEC and a Data DEC.

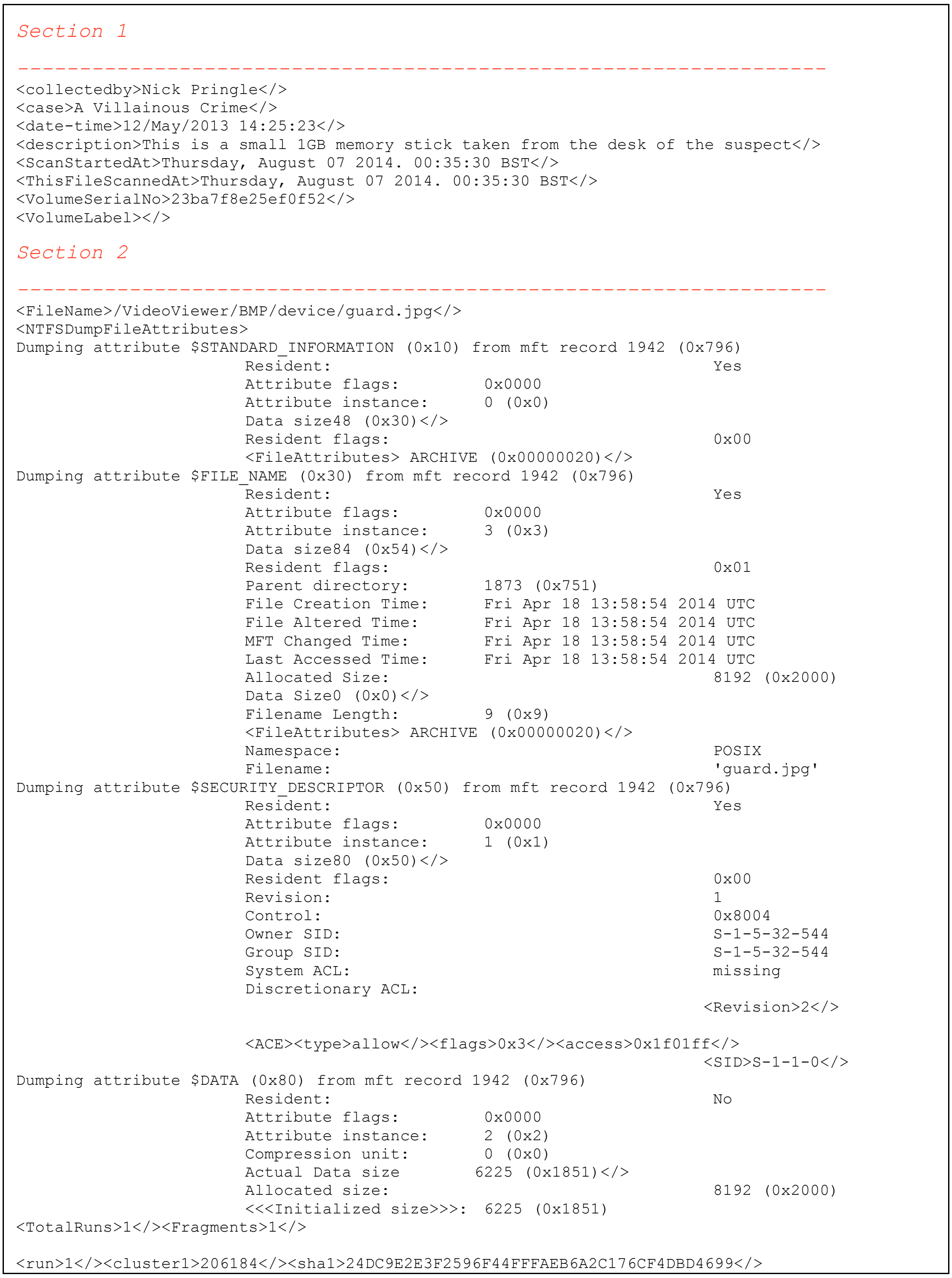
The Master DEC contains information about the file-system. It contains the first 20 characters of the key used to encrypt the data in the Data DECs, a reference to the scan data and time and a directory listing of all the files in the scan including the hidden system files.

```
<keytext>-NbtMSejN$c&LO^URH9</>
<scandatetime>2014-08-07  0:35:28 +00:00</>
<filesystemSerialNo>14A51B74C91B741C</>
[      4 -rwxrwxrwx        2560 May 11  2011]  ./$AttrDef
[      8 -rwxrwxrwx           0 May 11  2011]  ./$BadClus
[      6 -rwxrwxrwx       32736 May 11  2011]  ./$Bitmap
[      7 -rwxrwxrwx        8192 May 11  2011]  ./$Boot
[     11 -rwxrwxrwx           0 May 11  2011]  ./$Extend
[      2 -rwxrwxrwx     5361664 May 11  2011]  ./$LogFile
[      0 -rwxrwxrwx     2127872 May 11  2011]  ./$MFT
[      1 -rwxrwxrwx        4096 May 11  2011]  ./$MFTMirr
[      9 -rwxrwxrwx           0 May 11  2011]  ./$Secure
[     10 -rwxrwxrwx      131072 May 11  2011]  ./$UpCase
[      3 -rwxrwxrwx           0 May 11  2011]  ./$Volume
[   2044 drwxrwxrwx        4096 May 11  2011]  ./Acer
[   2027 -rwxrwxrwx       76800 Nov 30  2008]  ./A Grid GPS part 2.doc
[   2028 -rwxrwxrwx       11805 Nov 30  2008]  ./A Grid GPS.pdf
[   2029 -rwxrwxrwx      361472 Nov 29  2008]  ./A Proposal for a GPS Grid.ppt
..
..
[   1942 -rwxrwxrwx        6225 Aug  6  2008]  ./VideoViewer/BMP/device/guard.jpg
..
..
[   2041 -rwxrwxrwx    17331590 Mar 28  2011]  ./VideoViewer_Setup_M0178.exe
[   2025 drwxrwxrwx           0 May 14  2009]  ./Winhex
[   2026 -rwxrwxrwx     1246735 Jul  3  2008]  ./Winhex/winhex.zip
[   2042 -rwxrwxrwx     1373751 Jun 23  2009]  ./wrar39b3.exe

46 directories, 1967  files
```

Figure 88 - FCluster Master DEC

The Data DEC contains all the information needed to fully represent the data collected. There are three sections within the DEC. The First section contains header data about the

DEFR who actioned the scan, the case, the date and time the scan took place and information about the media and file system. The second section contains detailed meta-data about the data file. In the example in Figure 89 - FCluster Data DEC for a single file "`/VideoViewer/BMP/device/guard.jpg` ", we can see the meta-data for a file stored on an NTFS file-system. This includes details of the original cluster numbers and corresponding cryptographic hashes of the data from each of these clusters. In addition, a cryptographic hash of the entire file is stored. The third, and final section, contains a Uuencoded version of the original data encrypted using AES-256 with the key originally sent to the imaging device.



```
Section 1

------------------------------------------------------------------------------------
<collectedby>Nick Pringle</>
<case>A Villainous Crime</>
<date-time>12/May/2013 14:25:23</>
<description>This is a small 1GB memory stick taken from the desk of the suspect</>
<ScanStartedAt>Thursday, August 07 2014. 00:35:30 BST</>
<ThisFileScannedAt>Thursday, August 07 2014. 00:35:30 BST</>
<VolumeSerialNo>23ba7f8e25ef0f52</>
<VolumeLabel></>

Section 2

------------------------------------------------------------------------------------
<FileName>/VideoViewer/BMP/device/guard.jpg</>
<NTFSDumpFileAttributes>
Dumping attribute $STANDARD_INFORMATION (0x10) from mft record 1942 (0x796)
                        Resident:                                       Yes
                        Attribute flags:        0x0000
                        Attribute instance:     0 (0x0)
                        Data size48 (0x30)</>
                        Resident flags:                                 0x00
                        <FileAttributes> ARCHIVE (0x00000020)</>
Dumping attribute $FILE_NAME (0x30) from mft record 1942 (0x796)
                        Resident:                                       Yes
                        Attribute flags:        0x0000
                        Attribute instance:     3 (0x3)
                        Data size84 (0x54)</>
                        Resident flags:                                 0x01
                        Parent directory:       1873 (0x751)
                        File Creation Time:     Fri Apr 18 13:58:54 2014 UTC
                        File Altered Time:      Fri Apr 18 13:58:54 2014 UTC
                        MFT Changed Time:       Fri Apr 18 13:58:54 2014 UTC
                        Last Accessed Time:     Fri Apr 18 13:58:54 2014 UTC
                        Allocated Size:                                 8192 (0x2000)
                        Data Size0 (0x0)</>
                        Filename Length:        9 (0x9)
                        <FileAttributes> ARCHIVE (0x00000020)</>
                        Namespace:                                      POSIX
                        Filename:                                       'guard.jpg'
Dumping attribute $SECURITY_DESCRIPTOR (0x50) from mft record 1942 (0x796)
                        Resident:                                       Yes
                        Attribute flags:        0x0000
                        Attribute instance:     1 (0x1)
                        Data size80 (0x50)</>
                        Resident flags:                                 0x00
                        Revision:                                       1
                        Control:                                        0x8004
                        Owner SID:                                      S-1-5-32-544
                        Group SID:                                      S-1-5-32-544
                        System ACL:                                     missing
                        Discretionary ACL:
                                                                       <Revision>2</>

                        <ACE><type>allow</><flags>0x3</><access>0x1f01ff</>
                                                                       <SID>S-1-1-0</>
Dumping attribute $DATA (0x80) from mft record 1942 (0x796)
                        Resident:                                       No
                        Attribute flags:        0x0000
                        Attribute instance:     2 (0x2)
                        Compression unit:       0 (0x0)
                        Actual Data size       6225 (0x1851)</>
                        Allocated size:                                 8192 (0x2000)
                        <<<Initialized size>>>:  6225 (0x1851)
<TotalRuns>1</><Fragments>1</>

<run>1</><cluster1>206184</><sha1>24DC9E2E3F2596F44FFFAEB6A2C176CF4DBD4699</>
```

```
<run>1</><cluster2>206185</><sha1>E08556954353A3DECAD1E11524B12202C4593B7E</>
<WholeFileSHA1>FE3B0313FF0CA689B894DEFFF3978C410579253F</>
<NTFSInodeGeneralInfo>
<UpdSeqArrayOff>48 (0x30)</>
<UpdSeqArrayCount>3 (0x3)</>
<UpdSeqNumber>8 (0x8)</>
<LogFileSeqNumber>0x0</>
<MFTRecordSeqNumb>1 (0x1)</>
<NumberOfHardLinks>1 (0x1)</>
<AttributeOffset>56 (0x38)</>
<MFTRecordFlags>IN_USE </>
<BytesUsed>424 (0x1a8)></n>
<BytesAllocated>1024 (0x400)></>
<NextAttributeInstance>4 (0x4)</>
<MFTPadding>00 00 </>


Section 3
-------------------------------------------------------------------------
<data>
begin-base64 777 FE3B0313FF0CA689B894DEFFF3978C410579253F.cpt
rTDtUc7S3RR7Hu62FxOEtVdB0kFv0cwiqx59a4y9n+N3dz4qUr1+rbPRtXNW
AhZtYO/n7Krwglyu1O8SCNlpGZukwyyr99U0yDHIstmGckza8O/t63ZZR0ex
siP91GisjMI2FxBY7Bjc3J1nihN1i1qLwOz5wJu86pUG07Ij/7qmPhOoGEmt
..
..
..
D7yX2+AY1RcrfJECQd7bLnDexgrZPE0ad64Az1e2HsoNQ8+3bySis62xT545
f4PP7RvtDyrJ8BPy3qsY6kpKawWZbR3mUjJxQM6Y1RaEwyKVBWIUYWPSqeFQ
WT8=
====
</data>
```

Figure 89 - FCluster Data DEC

One key feature of the FCluster DEC is that given all the DECs it is possible to rebuild an exact reproduction of the original media accurate to the cluster location of data on the original.

DECs are named with a regular convention:

[File System Serial Number-[SHA-1 of the file contents].meta

```
14A51B74C91B741C-FE3B0313FF0CA689B894DEFFF3978C410579253F.meta
```

The FCluster DEC is not intended to progress into a fully working format. It is an expedient allowing more effort to be focused on the main issues with movement of data through the cluster.

6.6 Prioritising Data Acquisition

6.6.1 Introduction

Forensic analysis has become much more complicated in recent years. The term Velocity/Variety/Volume (3V), section 3.7, has been used to express this increase in data throughout the computer domain.

As investigators, we have no control over the media we are required to examine. It would be very useful if our investigations could be limited to evidence stored on small memory sticks or SSDs as these, by virtue of their capacity and speed, currently pose few challenges. One of our biggest problem areas is the analysis of the data found on large hard disks. Our techniques have not changed in more than a decade. We still take an entire image then copy it onto a workstation/server and initiate analysis. After this, we can start our manual analysis. This is linear.

In line with the principles addressed in section 4.5, we strive to introduce optimisation in the acquisition process. In this section, we introduce data processing prioritisation in which we use a Bayesian approach to a valuation of the potential. Using this 'self-assessment', every file is given a value that represents the investigator's feeling of the likelihood of evidence being found in the file. From this, we prioritise the sequence in which the data is acquired and so available for processing. By 'pushing' data, files with a high likelihood of yielding evidence towards the front of the processing queue we hope to make the process immediate and reduce 'noise'.

6.6.2 Current practice

6.6.2.1 Introduction

From our own experience, we have come to expect that a PC running Windows XP and with only one file-system partition, drive c:, when subjected to typical use after a couple of years, will have about 80,000 files. We have seen this increase to about 150,000 with Windows 7 and further increase to about 250,000 with Windows 8. Clearly, the number of files, their types and their sizes depends on the user but there is no doubt that volumes are increasing.

15 years ago, in 2000, when investigators were considering analysing the data on a 40GB drive running Windows XP they could expect acquisition times of about an hour and processing times of the order of 6 hours. This was not unreasonable given the time scale of investigations and the prevailing view, at the time, that digital information was of questionable provenance in an investigation.

If now takes about 24 hours just to read all the data from a 4TB hard disk, and perhaps many days, or even weeks, to process it, it is sensible to reassess the sequence in which we process the data.

Historically approaches to analysis have been largely sequential, perhaps influenced by the approaches to overall investigation management and evidence acquisition.

Forensic file formats were discussed in section 4.6.3. In this section, we revisit the same techniques but assess them from the point of view of speed of acquisition and access to the data for processing. We reviewed existing techniques for imaging in section 4.6.1. From that, we saw that there were two fundamental approaches to imaging

- Linear: where data is read from the source and replicated to the destination in a strictly linear sequence, normally by media block sequence;
- Non-linear: where some strategy is adopted with the intension of maximising the time efficiency of the task.

6.6.2.2 The linear approach - by Media Cluster

This has been the prevailing method for the last 20 years. Derived from the use of the ‘dd’ utility in *ix operating systems, it ignores any structures on the media and reads the data at block level. See section 4.6.2.2

This has one key advantage; because it is linear, it runs as fast as the source media will deliver data and the destination media can write.

Linear imaging has two possible outputs. It could be (a) as a file within a file system or (b) as an identical replica on another media device where the data is written directly to the media block structure.

a) `dd if=/dev/sda of=/home/forensics/myimagefile`

b) `dd if=/dev/sda of=/dev/sdb`

The former has an advantage because the process could deliver the result straight to a storage device available to the processing system e.g. a file server accessible via a network. The latter would normally need to be recopied onto the file storage on the processing system.

Linear imaging has a major drawback in that the result cannot be accessed for processing until the entire imaging process is complete.

6.6.2.3 Non-linear Approaches

Adopting a Digital Evidence Container or Submission Information Package format during acquisition, allows us to take an alternative approach by parsing the data and prioritising data made available for processing. This could include a selection and/or sorting criterion where we can reduce the overall capture. There is, however, a danger in selecting data, as opposed to collecting it all. It may be that key evidence is somewhere we never anticipated. This technique relies on the quality of informed focus.

In sections 4.6.2.6, 4.6.2.7 and 4.6.2.8, we described several evidence container formats that can hold individual files rather than whole images. In building the former, there are a number of ways data could be sorted and selected for inclusion.

6.6.2.3.1 By using the name of a Folder or File name

Files could be sorted and selected based upon folder and filename. For example, on windows XP the user files are held under “/Documents and Settings” and so these would be processed before “/Program Files” and “/Windows". On Windows 7 system this would put “/Users” before “/Windows”. In Linux, we would choose “/home” before “/etc”. Within a top-level folder like “/User” there could be many users and within them many sub-folders, some of which are setup by the system or applications programs, some are used by the user to store their data, for example “My Documents”. The problem with this is that the arrangement of files and folders varies so much.

6.6.2.3.2 By inode number

In most file systems, filenames, and their associated meta-data details, are stored in a file-system’s database (in NTFS, Ext and HFS) as a series of records, one per directory entry. These records are filled sequentially as files are added to the file-system. This means that, at least initially, as the operating system is the first thing written to the file-system, the operating system files will occupy the earlier records. Most likely, as the user installs initial application programs, next files are the application programs and associated files. These therefore occupy the next series of records and the user files are last to be created and added. When a file is deleted, the record it once occupied is marked as available and can be reused when a new file is created.

As the first files stored on a disk are those that comprise the operating system and application programs, there is a tendency for user files, which are the most likely source of evidence, to be placed towards the end of the file-system inode list. If we processed in inode sequence order, we are delaying the processing of the evidence rich data until last. This does offer a strong argument for a reverse inode sequence processing.

6.6.2.3.3 By File Size

The increase in media size and the connectivity of our digital devices over the last few years has resulted in changes in the way we use storage media. Prior to the mid-1990s, local storage, and local storage was the only type available, was where we stored our own work. During the mid-1990s, as Internet access became more common and so we increasingly used our larger local storage to keep our copy of data sourced from the Internet. Arguably, we now have greater cheap storage than we will ever need for our personal data storage needs. We suggest that massive media, currently of 2 TB and above, remains largely empty with a few exceptions. Copies of studio produced ‘feature’ films, *youtube* type downloads and originals of ‘home movies’ are often found occupying large media.

Our research, in the appendix, has indicated that a high percentage of files within Microsoft Windows are less than 10k in size.

In the form of statements, we found:

- There are a high percentage of files with very small sizes. E.g. 62% of 325,372 files are < 10k;
- There are a low number of very large files. 29 files greater than 1GB totalling 168GB.

It is disproportionately time consuming to process many small files as opposed to a few files that total the same data volume. This is because of the latency of opening and closing files is disproportionately high when compared to processing the contents. Conversely, it can be argued that processing a small number of very large files is a high-risk strategy where a large amount of processing power and time could be spent to yield not results. It may be better to process 1 x 1MB file rather than 100 x 10kB files.

6.6.2.3.4 By type

The obvious analogue in conventional forensics it one where the forensic team might look for finger prints on doors, windows, light switches and ‘the murder weapon’. We could focus on specific file types. For example, we could have a search sequence of:

*.doc
*.xls
Index.dat
*.jpg
*.pst

This approach is likely to be efficient but only if the selection were appropriate for the specific investigation in that in the case involving pornographic images would select JPGs before XLSs and a case of fraud would choose XLSs before JPGs. It would also need to be sensitive not just to the existence of a file but the implications of placing it in a processing queue ahead of other files. In addition, there may be huge numbers of one type of file. For example, it is not uncommon for an enthusiastic photographer to have tens of thousands of JPGs. This may require further refinement within the sequence that the JPGs are processed.

6.6.2.3.5 By Date

All files have a 'Last Modified Date' that can be used to isolate them on a time line. In some cases, this is not of use, for example, Outlook PST file will have a 'Last Modified Date' but it yields no information about the dates and times associated with each of the emails recorded within the file.

6.6.2.3.6 By Processing time

We could choose to process 'low hanging fruit' to yield quick results. This may be, for example, to extract EXIF information first from all JPGs rather than to attempt any face recognition on the files. Conversely, we could avoid creating thumbnails of key frames in a massive video file until later.

6.6.3 Bayesian Allocation of values

We allow values to be assigned to the following parameters

	File Attribute	Description
1	By Folder or Filename	Specific files or the contents of entire folders can be allocated scores to push them up or down the ratings.
2	By inode	We could favour files, that were generally (see section 6.6.2.3.2 for limitations) created earlier or later in the history of the file-system.
3	Size of the file: in MB	This could be used to exclude small files and the latency overhead associated with them.
4	Files by Extension (type)	: 1 – 9, representing the likelihood of evidence being present. e.g. JPG images are likely to be good sources of evidence, hence score 9, whereas .pg5 "Guitar pro 5 Tablature" Files are not and so would score 1.
5	File Extension Category.	File types can be grouped into categories such as Video, Text, spreadsheet rather than specific file extensions. Giving these groups a value can speed allocation of parameters.

6	Last Modified Date	Many crimes can be reduced to having been taken place within a certain period. This would most likely be an array of values for discrete periods, with 9 representing periods of high interest and 1 representing periods of low interest.
		`Jan  - 2013 - 3`
		`Feb  - 2013 - 7`
		`Mar  - 2013 - 9`
		`Apr  - 2013 - 5`
		`May  - 2013 - 3`
		`Jun  - 2013 - 9`
		`July - 2013 - 1`
		Continuously accessed files like logs could not be classified in this way and so could be allocated 9 as being always of interest, or 1 as never.

	File Attribute	Description
7	KFF	If this were known as 'ok' in the NIST database, it would score -10 to push it down the score table. However, its presence in the CEOP database would attract a score of 10 to push it up. The problem with this characteristic is that the entire contents of the file have to be processed to create the hash. By that time, it may as well have been processed.
8	By exceptional location in the file system	A JPG in any location other than "My Documents" is 'normal' and so could be set at 0, however the same file in "/Windows/System32" is abnormal and so could be score 7.
9	Processing rate: for this type of data on our equipment	Our experience tells us that processing it is possible to process 2 MB/s on average.

Table 14 - Evidence Value Score Parameters

Ultimately, we aim to create an index number for each file.

Index =

(K_1 * Filename or Folder Value)

+ (K_2 * inode)

+ (K_3 * File Size)

+ (K_4 * Ext type Value)

+ (K_5 * Category Value)

+ (K_6 * Date index)

+ (K_7 * Known File Fingerprinting index)

+ (K_8 * Location normality)

+ (K_9 * Size / Processing Rate)

We believe the constants (K_1..K_9) would need to be adjusted depending on feedback from real cases, as Horsman, Laing and Vickers (2014) suggest.

If Known File Fingerprint lookup is used to either eliminate known good files or highlight known bad files, it should effect the seventh term so that it either forcing inclusion or exclusion. Eg K_7 = 1000 or K_7 = 0.

Prioritisation will be evaluated in section 8.4.

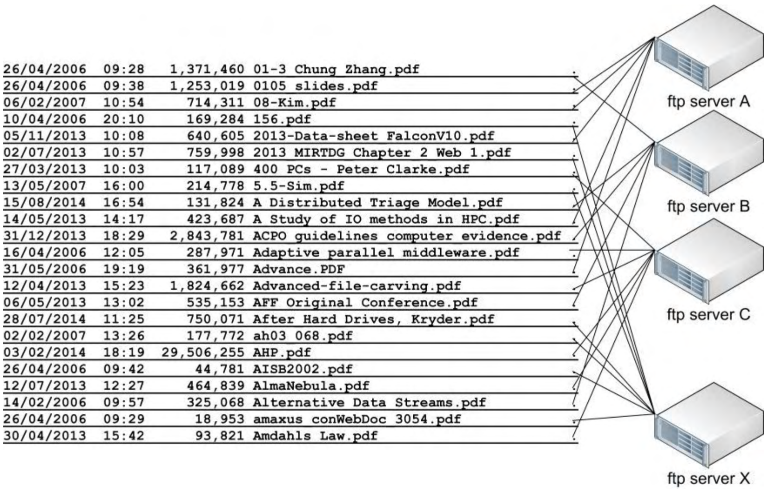
6.7 FClusterfs

6.7.1 Introduction

In chapter 5, we set out a number of design objectives for FCluster. These included the ability to store file data across a distributed cluster, store the associated file meta-data, have highly tailored access control and have the ability to track access and processing. As discussed, these are very similar to the basic functions of a file system.

We have observed that Hadoop employs a custom designed file system, HDFS, as a base for its processing, see section 4.3.1.3. HDFS has been implemented as a middleware on top of the native file-system used by the operating system running on the hosts within the cluster and below the application layer. Basing assurance upon a custom designed file system means that a layer is formed between any application programs and the data upon which they operate, which could not be circumvented. We have chosen to call this file –system FClusterfs.

Data storage in the FClusterfs file-system is dispersed across the hosts of the cluster. Whereas an NTFS file-system would have pointers to the local cluster number at which the data would start within its partition, FClusterfs' meta-data includes a pointer to a remote storage server where the entire file is located. This is not to be confused with mounting a network drive that mounts a whole directory tree on a mount point within the local file-system. In FClusterfs, the data of files, whose entries are adjacent within a directory listing, may not be within the same file-system partition as each other or even the same host machine. Figure 90 shows how files adjacent in a directory listing can be on separate servers using the native file system such as ext3/4.



```
26/04/2006  09:28   1,371,460 01-3 Chung Zhang.pdf
26/04/2006  09:38   1,253,019 0105 slides.pdf
06/02/2007  10:54     714,311 08-Kim.pdf
10/04/2006  20:10     169,284 156.pdf
05/11/2013  10:08     640,605 2013-Data-sheet FalconV10.pdf
02/07/2013  10:57     759,998 2013 MIRTDG Chapter 2 Web 1.pdf
27/03/2013  10:03     117,089 400 PCs - Peter Clarke.pdf
13/05/2007  16:00     214,778 5.5-Sim.pdf
15/08/2014  16:54     131,824 A Distributed Triage Model.pdf
14/05/2013  14:17     423,687 A Study of IO methods in HPC.pdf
31/12/2013  18:29   2,843,781 ACPO guidelines computer evidence.pdf
16/04/2006  12:05     287,971 Adaptive parallel middleware.pdf
31/05/2006  19:19     361,977 Advance.PDF
12/04/2013  15:23   1,824,662 Advanced-file-carving.pdf
06/05/2013  13:02     535,153 AFF Original Conference.pdf
28/07/2014  11:25     750,071 After Hard Drives, Kryder.pdf
02/02/2007  13:26     177,772 ah03 068.pdf
03/02/2014  18:19  29,506,255 AHP.pdf
26/04/2006  09:42      44,781 AISB2002.pdf
12/07/2013  12:27     464,839 AlmaNebula.pdf
14/02/2006  09:57     325,068 Alternative Data Streams.pdf
26/04/2006  09:29      18,953 amaxus conWebDoc 3054.pdf
30/04/2013  15:42      93,821 Amdahls Law.pdf
```

Figure 90 – Fclusterfs - Individual Files on individual servers

FClusterfs extends the 'normal' metadata found in file-systems such as NTFS and EXT3/4, which describe the file characteristics of the file within the storage server's file-system, to include information needed to justify the claim of assurance in Chain of Evidence. This includes information about the original source media, the date and time it was acquired, the date and time it was ingested into the cluster, where it was stored, and who and when it was access for processing. It supports distribution by recording the data about the server upon which the actual file data is stored within the cluster so that subsequently messages can be sent to the server, via the database, to initiate processing.

6.7.2 FClusterfs implemented as a FUSE File System

6.7.2.1 FUSE File-Systems in Digital Forensics

The use of FUSE to build custom file systems has already been proposed within the digital forensics domain (Richard et al. 2007). FClusterfs advances this notion and uses the technique to provide a solution that addresses the key issues of Assurance in a distributed processing environment. It merges the functionality of several existing FUSE file systems, then adds and extends them to form a new file system.

FClusterfs is a FUSE file-system that holds the file-system metadata, normally held in the native media file system, in a MySQL database while its data is spread across remote storage servers.

6.7.2.2 FClusterfs is based on MySQLfs

FClusterfs is based on MySQLfs (Brancatelli 2014). MySQLfs employs an SQL database consisting of three tables to replace the native file system. The ‘inodes’ table provides storage for file metadata like names, dates/times, size, access rights etc usually seen as a ‘directory’. The ‘tree’ table stores the hierarchical structure of folders and filenames found in the file-system. The third table ‘data_blocks’ stores the actual data as a series of binary large objects (BLOBs), replacing the clusters of the disk format. More details can be found in 4.5.2 .

In FClusterfs, we use the tree and inodes tables found in MySQLfs and have added a table called ‘meta-data’ to store the additional meta-data from the original location of the data. This is a variable length, large text field and so is better in a table of its own rather than extending the inodes table.

6.7.2.3 Multivolume within a Single Database

A single FClusterfs database is designed to store many file-systems. We have a table, VolumeInformation, which contains a record of each file-system whose individual records are stored within the inodes table. We have added a field ‘VolumeID’ to the inodes table, as a back reference, to identify the source file-system. This is not technically required within a relational database, as MySQL conforms closely to the ACID model (MySQL 2014), but has been added to raise assurance.

6.7.2.4 Remote storage

We substitute the functionality of the ‘data_blocks’ table in MySQLfs with the ability to read data stored on remote storage servers.

In the prototype, proof of concept design, we have chosen to connect to the remote servers using the ftp protocol because of the features of another existing FUSE file-system curlFTPfs (Robson 2013). curlFTPfs allows the user to mount a connection to an ftp server and make it appear to be part of the host’s file system. curlFTPfs attains much of its power and flexibility because it is based in the libcurl library and can support not only ftp but SSH, SFTP, HTTP, HTTPS but, despite obvious security issues with unencrypted data transfer, we have chosen ftp as a simple base for a prototype as it avoids the complexities of dealing with cryptographic keys. In a real world scenario, SSH would be a more robust choice.

curlFTPfs allows only one storage server per mounted file system. In FClusterfs, we have enhanced this to be able to access individual files on any storage server on a file-by-file basis. The corresponding server details are stored in the individual file’s record in additional fields that have added to the ‘inodes’ table. When the user sees a directory listing in their user space it appears as a continuous list drawn from the ‘inodes’ table but in reality each file’s data will be on a storage server which is most likely remote. Each file is held in its entirety on the storage server. In the prototype, the entire file is transferred and held in cache in memory. In curlFTPfs, 128MB chunks are transferred just once and, if the file is over 128MB, a mosaic is built in a cache in local memory. In FClusterfs, unlike many distributed file-systems, file data is stored as a complete file on the storage server. It is not segmented or striped. This enables another key feature of FCluster to be implemented with ease; local processing.

6.7.2.5 Primary focus on Local Access for Processing

It is important to realise that although FClusterfs does allow data to be transported across the network, it is primarily a means of standardising access to data held locally. When we use the term ‘passes across the network’ it should be normally taken as via 127.0.0.1, the localhost loopback connection (Figure 91 – File Access via 127.0.0.1).

FClusterfs is intended to replace any need for network shares like NFS or SMB but to emphasise, although FCluster provides access to the file-system under investigation and will work over a local or wide area network connection but is inefficient. It is intended to process local data by the host of the ftp server holding each of the files. The location, URL, of the storage server hosting the data is part of the ‘inodes’ table extending the fields used by FClusterfs and so the ‘locality’ of the file can trigger the processing task to be initiated within the host.

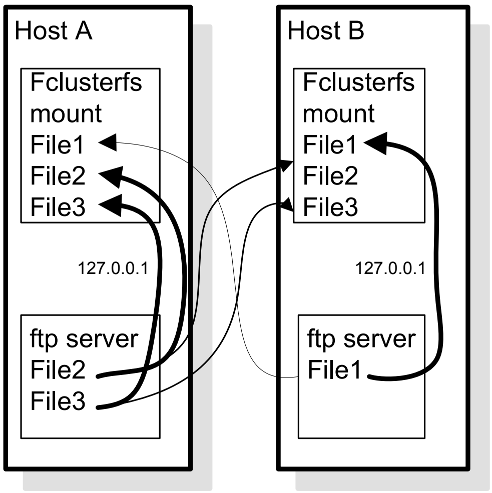


Figure 91 – File Access via 127.0.0.1

In Figure 91 – File Access via 127.0.0.1, we show two host computers each with an ftp server that stores the files that make up a simple directory held in FClusterfs. Although each host sees the complete directory listing, we can see the files are split across the two servers. FCluster would normally try to restrict processing of File2 and File 3 to hosts A and File1 to host B. This avoids any transfer of data across the network. Local data can be accessed via either the 127.0.0.1 localhost or the local IP number, as routing will prevent it needing to cross the network media. This does not prevent remote access to data.

6.7.2.6 Encryption

Data held on this multitude of ftp servers is encrypted and uses techniques from ecryptfs (Hicks et al. 2013) to decrypt data on-the-fly. After it leaves the media of the ftp server, it passes across the network and is decrypted in the user's host RAM before being held in cached space in RAM in their Virtual File System.

6.7.2.7 Read Only Access

As previously mentioned, FClusterfs provides read-only access. There is no code to implement functions like write / delete / chown / chmod. This is a fundamental requirement of a forensic system and, fortuitously, greatly simplifies the code and enhances the speed of FClusterfs. This based upon ROfs (Keller 2014).

6.7.2.8 Auditing

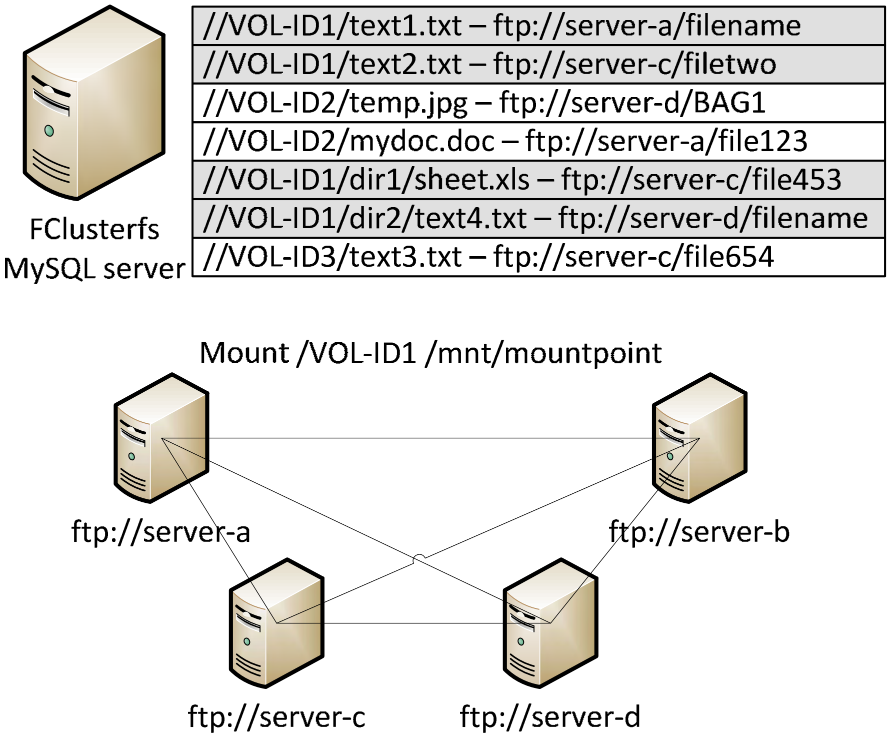
FCluster has auditing which it draws from Loggedfs (Flament 2013). Loggedfs' audit is felt to be too granular for our purposes and instead we choose to record only significant actions like DEC movement, unpacking and the opening of data-files for processing. FClusterfs records an entry every time the 'Open' function is called. Recording access to parts of a file is felt to be unnecessary and would only slow the system and make the logs unreadable. All audit records are stored in a table ‘audit’ recording date/times, users. The audit entries for DEC movement, unpacking etc are actioned within the daemons that control these activities.

6.7.2.9 Storage Server Access Information

Although the data location URL information is available to the user e.g. ftp://myserver.com/, the username and password needed to log into the storage server and gain access the data is not. It is held in another table ‘serveraccessinfo’ and is retrieved on the fly during a read request by FClusterfs. Users can only access evidence via the FClusterfs file-system that provides data from the storage servers. Users never know the credentials needed to access data on the storage servers. This is held within the FClusterfs MySQL database can only be accessed by the daemons that control data movement etc. In a real world implementation remote data access would not use ftp and would not use plain text credentials. Instead, perhaps SSH would be used together with PKI authentication.

6.7.2.10 Peer to Peer

FClusterfs is also peer-to-peer and so any node can mount a directory that can reference files on any server (Figure 92 - FCluster mounts peer to peer).



//VOL-ID1/text1.txt – ftp://server-a/filename
//VOL-ID1/text2.txt – ftp://server-c/filetwo
//VOL-ID2/temp.jpg – ftp://server-d/BAG1
//VOL-ID2/mydoc.doc – ftp://server-a/file123
//VOL-ID1/dir1/sheet.xls – ftp://server-c/file453
//VOL-ID1/dir2/text4.txt – ftp://server-d/filename
//VOL-ID3/text3.txt – ftp://server-c/file654

Figure 92 - FCluster mounts peer to peer

6.7.2.11 Mounting the FCluster file-system

The behaviour of an FClusterfs file system is defined when it is mounted by a command line that contains the following entries:

```
fclusterfs                                                              \
 --mysql_user=me                                                        \
 --mysql_password=mypassword                                            \
--mysql_host=25.63.133.244                                              \
 --mysql_database=fclusterfs                                            \
--volume=74a8f0f627cc0dc6                                               \
 --audituser='Investigator                    Name'                     \
 /home/user/Desktop/fsmount
```

Multiple file systems can be mounted on the user's host system and multiple SQL servers can provide storage for FClusterfs file-system databases.

FUSE file systems work in the User's data-space. File-systems mounted by one user, under their authorisation, are not visible to any other users.

6.7.2.12 Storage of File Meta-data

Jigsaw imaging creates DECs that contain data from the original source context. This is travels with the DEC and is unpacked when the data is ingested into FCluster. All meta-data for files is held in the MySQL database in the meta-data table. In the proto-type, the data is held as a 'text' field and the meta-data is delimited with XML formatting. In future developments, some meta-data might be extracted and stored in its own fixed format fields to enable indexing and selecting.

6.7.2.13 Cryptographic hashes meta-data checking

One such item is the cryptographic hash created at imaging time. This is so important that it is extracted and stored in its own fixed length field in the inodes table for indexing and retrieval.

6.7.2.14 Accessing data with a legacy program

FClusterfs enables files to be mounted on an existing file system as if they were 'normal' files. The application layer programs are completely unaware of the actions of FClusterfs below it. This is specifically what FUSE file systems do. Subsequently, subject to access controls with FClusterfs and the speed of the network connection between the application program and the data storage media, legacy program are completely unaware of the environment in which they operate with regards files storage. As FUSE file systems are stackable, these mounts can, in turn, be mounted and so become available to software hosted on other operating systems. For example, FTK, Encase or Bulk Extractor running on

Windows would be unaware that the data that presents as begin on a CIF, SMB network share is actually hosted on a system running FCluster and FClusterfs.

6.7.3 The MYSQL database at the heart of FClusterfs

6.7.3.1 Introduction

FClusterfs is based on is the MySQL database inherited from MySQLfs. Data is held as a hierarchical structure (Figure 93 - FClusterfs Hierarchy).

- FCluster may have one or more hosts;
- An FCluster host may have none, one or more than one MySQL servers;
- An FCluster system must have one of more MySQL servers to support FClusterfs;
- A MySQL server may have one or more FClusterfs Databases;
- An FClusterfs Database may have one or more file-systems within it;
- An FCluster host may have none, none or more storage servers.

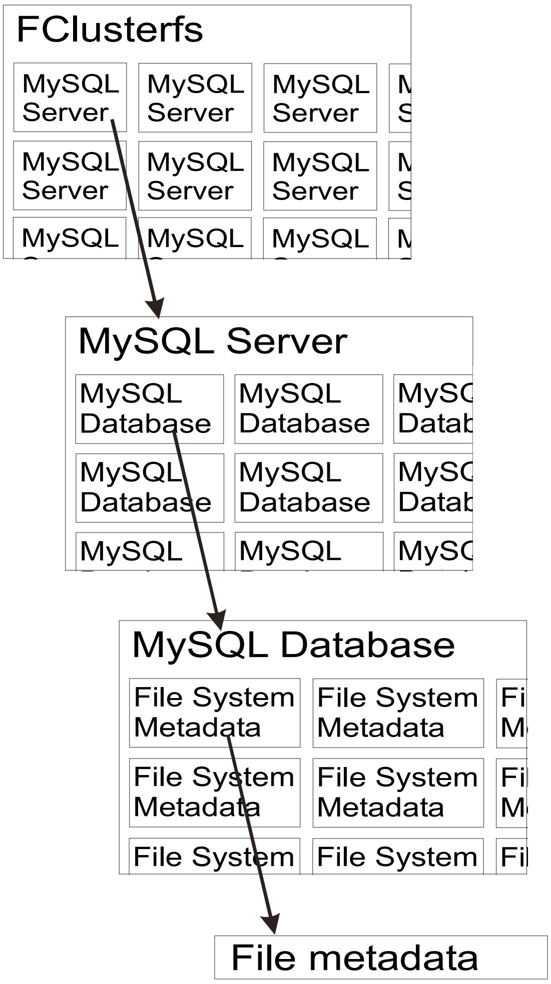


Figure 93 - FClusterfs Hierarchy

6.7.3.2 Structure of Database

An FClusterfs database consists of nine tables shown in Table 15

	Table Name	Description
#1	Devicelisting	This table holds a list of all the imaging devices acknowledged by the system. It holds details about serial number, device testing/calibration history
#2	VolumeListing	A table holding details of cryptographic keys generated and allocated to a specific device. These keys have a generation date/time and an expiry date/time. When the DEC data is ingested this table is updated with the file-system volume identifier, scan date and base record ID of the directory system in the tree, inodes and meta tables
#3	Inodes	A table holding data about every single data file held in the system. In addition to data held by the evidence file system, extra data is held about evidence cryptographic checksums, data locations on the FCluster etc.
#4	Tree	This is where the hierarchical structure of the data files acquired when imaging is stored. This is populated as the DECs are imported.
#5	MetaData	This holds the meta data about the evidence file gathered at imaging acquisition time. This includes the cluster numbers the file occupied on the original media and individual cryptographic checksums for each cluster.
#6	Serveraccessinfo	Contains the access information (user login, password, server type ftp, ssh, https etc) of all the datanodes on the FCluster.
#7	Audit	Contains records of activities. These include data movement, unpacking, opening and deletion.
#8	WorkFlow Tasks	Contain a series of tasks that are routinely initiated when data of a certain type arrives on a storage server.
#9	NodeState	A record of the characteristics of the hosts that comprise the cluster including storage capacity, capability metrics that represent the processing power of the host and network connection metrics.

Table 15 - List of Tables in MySQL

6.7.3.3 *Structure of Tables*

6.7.3.3.1 DeviceListing table

In contrast to existing systems, imaging devices and software are seen as part of the FCluster system. As such, we keep a record of their existence for validation and to grant authority to image, see 6.3.4 Information about these devices is held in the DeviceListing table, see Table 16

Field Name	Typical Data Type	Description
ID	BigInt(20)	Unique ID for Indexing
Description	Varchar(255)	Description of Device
Custodian	Varchar(100)	The person responsible for this equipment
Last Tested	Datetime	Last test/calibration data
Serial Number	Varchar(50)	Serial Number
Narrative	longtext	Description

Table 16 - DeviceListing table structure

6.7.3.3.2 VolumeListing table

To enable multiple source file-system volumes to be stored within one database, information about each source volume is stored within the VolumeListing table in Table 17.

Field Name	Typical Data Type	Description
ID	BigInt(20)	Unique ID for Indexing
DeviceID	BigInt(20)	Device reference for the Imagine Device
IssuedDateTime	DateTime	Date Time this Authority to Image was created
ExpiresDateTime	DateTime	Date Time this Authority will Expire
ScanDateTime	DateTime	Date Time the Scan took place
FSRootInode	BigInt(20)	The ID number within the Inodes table, for the root of this file system tree
Keytext	Varchar(1024)	The random string used to encrypt the Digital Evidence Containers in this scan

Table 17 - VolumeListing table structure

6.7.3.3.3 Inodes table

The Inodes table replaces the directory data structure in a native file system, for example $MFT in NTFS. It is the main store for all the directory information taken from the original source media and is extended to include the extra meta-data required in a distributed environment. Most of the entries in inodes are expected to represent files but some will represent directory entries, in which case the storage server meta-data is not necessary. Table 18 lists the fields in the prototype implementation.

Field Name	Type and Size	Description
Inode	BigInt(20)	Unique ID for the entry
VolumeID	Char(45)	ID in the VolumeListing table
FSFilename	Varchar(255)	filename
Uid	VarChar(50)	Owner information from the original file-system
Gid	VarChar(50)	Group owner information from the original file-system
Atime	DateTime	Last Access Date/Time from the original file-system (Read-only)
Mtime	DateTime	Last Modified Date/Time from the original file-system (Read-only)
Ctime	DateTime	Creation Date/Time from the original file-system (Read-only)
Size	BigInt(20)	Size of the file
SHA1	VarChar(40)	Cryptographic Hash
Originallocation	Varchar(1024)	Full path name on the original file system
Firststorageprotcol	VarChar(10)	Eg ftp, ssh, http
Firststorageserver	VarChar(40)	IP No or name of storage server
Firststorageinplace	VarChar(1)	True if the DEC has been copied to the primary storage server
Firststorearrivaldatetime	DateTime	Date/Time of the arrival of the DEC on the primary storage server
Firststorageunpacked	DateTime	True if the DEC have been unpacked (and so is available for processing)
Firststoragelastvalidated	DateTime	Date/Time the replication daemon last tested the integrity of the data
Firststorageprocessed	DateTime	True if the standard work tasks have been completed on the data

Field Name	Type and Size	Description
Secondstorageprotcol	VarChar(10)	See description as First storage
Secondstorageserver	VarChar(40)	..
Secondstorageinplace	VarChar(1)	..
Secondstoragearrivaldatetime	DateTime	..
Secondstorageunpacked	DateTime	..
Secondstoragelastvalidated	DateTime	..
Secondstorageprocessed	DateTime	..
Thirdstorageprotcol	VarChar(10)	..
Thirdstorageserver	VarChar(40)	..
Thirdstorageinplace	VarChar(1)	..
Thirdstoragearrivaldatetime	DateTime	..
Thirdstorageunpacked	DateTime	..
Thirdstoragelastvalidated	DateTime	..
Thirdstorageprocessed	DateTime	..

Table 18 - inodes table structure

6.7.3.3.4 Tree table

The tree table stores the hierarchical tree structure from the original media. See Table 19 - tree table structure

Field Name	Type and Size	Description
Inode	BigInt(20)	Unique ID for corresponding entry in the inodes table
VolumeID	Char(45)	Unique Volume ID for entry in VolumeListingID table
Parent	BigInt(20)	Parent inode in file-system hierachy
name	Varchar(255)	File/Directory name

Table 19 - tree table structure

6.7.3.3.5 MetaData table

The Metadata table is used to store the meta-data from the original media. This data is created when the evidence is acquired; see 6.4.2, The Jigsaw Technique and forms part of the data stored in the DECs. It is unpacked when the DEC arrives on the storage server. In addition, the results of data processing are recorded in the metadata table. See Table 20 - metadata table structure.

Field Name	Type and Size	Description
Inode	Bigint(20)	Unique ID for corresponding entry in the inodes table
Metadata	Longtext	Text field for data gathered at acquisition time
VolumeID	Varchar(45)	Unique Volume ID for entry in VolumeListingID table
Result1	Longtext	Text result of worktask processing. Could just be “Success”
Result2	Longtext	Text result of worktask processing. Could just be “Success”

Table 20 - metadata table structure

6.7.3.3.6 Audit table

Every significant action upon the data is recorded in the audit table. These include movements, opening the file. This includes action initiated by the user but also actions initiated by the FCluster control Daemons and automated workflow tasks. The structure can be seen in Table 21 - audit table structure.

Field Name	Type and Size	Description
ID	BigInt(20)	Unique ID for record
DateTime	Char(40)	Data/Time for entry
Investigator	Char(45)	User ID for agent of action. Could be investigator or daemon name
Action	Char(45)	Description
Inode	Bigint(20)	Unique ID for corresponding entry in the inodes table
VolumeID	Varchar(45)	Unique Volume ID for entry in VolumeListingID table

Table 21 - audit table structure

6.7.3.3.7 WorkFlowTasks table

As data arrives to each of the data storage servers this provision for a set of ‘standard’ tasks to be completed on the data. This could be, for example, extracting the EXIF data from JPGs or extracting data from an Outlook PST file. Ad-hoc processing, initiated by the user is also added to this table which forms a job queue to be executed as and when the processor has time. This task information is held in the workflow table, Table 22 - workflow table structure.

Field Name	Type and Size	Description
FileExtension	Varchar(10)	Eg *.jpg, *.doc to describe the file types to which this applies
Priority	Int(11)	To enable priority control
ExecutableName	Varchar(255)	CLI command line

Table 22 - workflow table structure

6.7.3.3.8 ServerAccessInfo table

The access credentials for each storage server is stored within FClusterfs. Multiple storage hosts e.g. ftp, ssh, http could exist on a host and so multiple entries can be made for a host. Table 23 - serveraccessinfo table structure

Field Name	Type and Size	Description
ID	Int(11)	Unique ID for indexing
Protocol	Varchar(45)	Communications protocol eg ftp, ssh, http
IP	Varchar(45)	IP No or hostname
Port	Int(5)	Port on which this server is listening
User	Varchar(45)	Format user:password

Table 23 - serveraccessinfo table structure

6.7.3.3.9 NodeState table

The data in the NodeState table is used to make decisions when allocating data out to the nodes for storage and processing. See 6.3.4.3.1

Field Name	Type and Size	Description
ID	Bigint(20)	Unique ID for indexing
Hostname	Varchar(20)	Hostname
LastUpdate	DateTime	Date/time of last data update
IPNo	Varchar(20)	IP number of host
CPUIdle0	Float	% idle time of this core
CPUIdle1	Float	..
CPUIdle2	Float	..
CPUIdle3	Float	..
CPUIdle4	Float	..
CPUIdle5	Float	..
CPUIdle6	Float	..
CPUIdle7	Float	..
StorageRoot	Varchar(200)	Full path to the root of the local storage allocation
StorageAvailable	Bigint(20)	Storage capacity allocated for use on the storage host
StorageUsed	Bigint(20)	Storage used on the storage host.
ProcessorIndex	Bigint(20)	Processor Speed Index
MediaIndex	Bigint(20)	Storage Speed Index
LinkSpeed	Bigint(20)	Effective Link Speed from this database host.

Table 24 - nodestate table structure

6.7.3.4 Table relationships

The relationship between the important tables is shown in Figure 94. This is mostly the link between inode, tree and metadata by the common information about the file's inode.

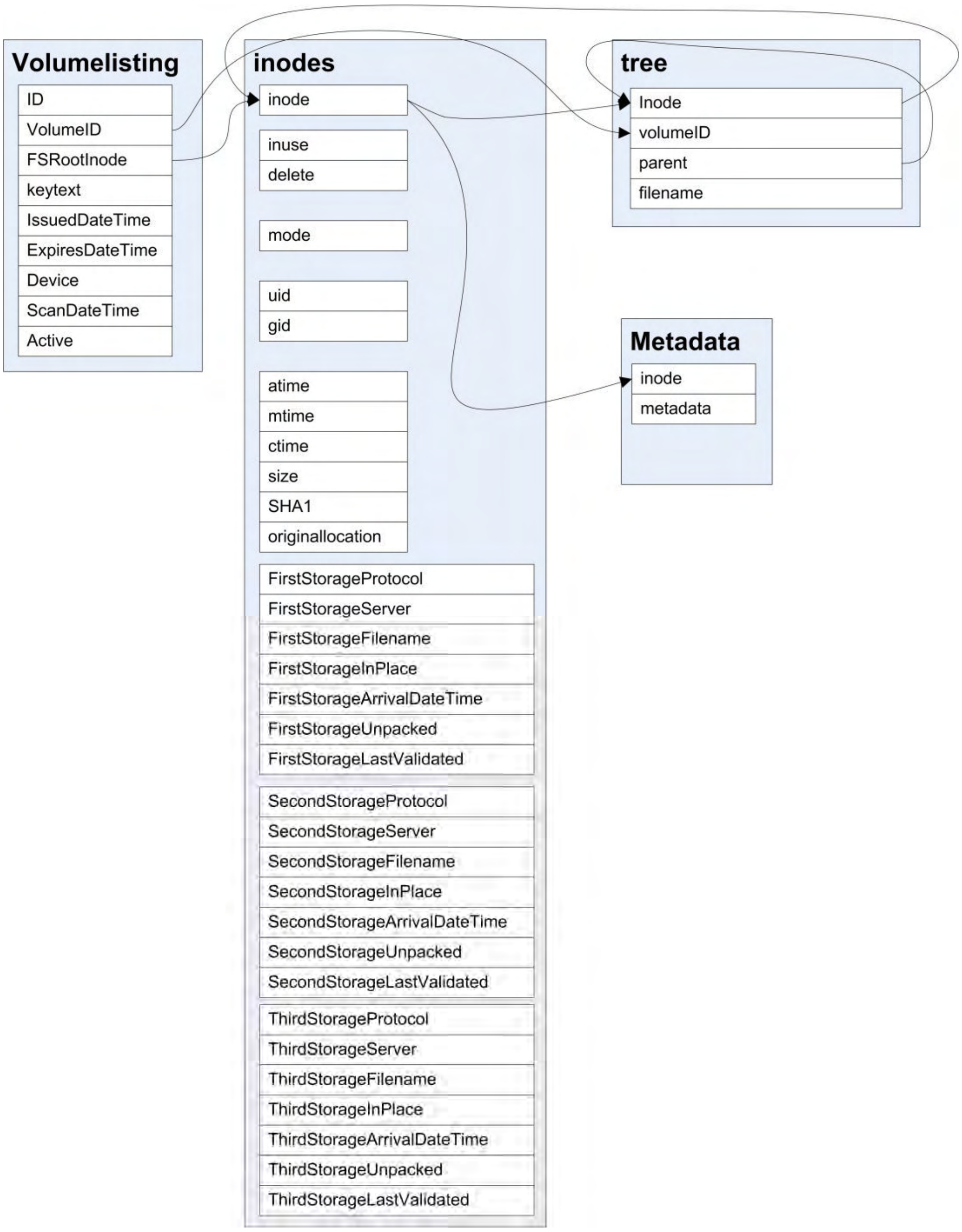


Figure 94 - FClusterfs Tables

6.7.3.5 The use of tables within assurance zones

Table 25 shows which table is used within each assurance zone.

Sub System	Database table	Program
Acquisition	Device Listing VolumeListing Audit	Acquisition Authority
Ingestion	VolumeListing Inodes Tree Audit	Load Balancer
Distribution	NodeState Inodes Tree Metadata Serveraccessinfo Audit	Movefiles unpackfiles
Workflow Processing	VolumeListing Inodes Tree Metadata Workflowtasks Audit	Jobprocessing mysqlftpfs
Adhoc Processing	VolumeListing Inodes Tree Metadata Audit	mysqlftpfs dojob

Table 25 - The use of tables with specific Subsystems

6.7.3.6 Fields updated in the inodes table as data progresses through the cluster

As DECs/SIPs/AIPs progress through ingestion into FCluster a series of fields within the inodes table are updated, Table 26, on the following page, shows the progress through each stage of ingestion. Progress through the assurance zones is dependent on the data for the previous assurance zone being present in the inodes table.

Progress of data through the cluster >>>>>>>>>>							
Field Name updated in the inodes table	Updated at Ingestion stage 1	Updated at Ingestion Data Import	Updated at Ingestion Primary Allocation	Updated at Ingestion Replication	Updated at Distribution and unpacking	Updated after Primary Processing	Updated after Routine Validation
Inode	X						
VolumeID	X						
Fsfilename	X						
Uid	X						
Gid	X						
Atime		X					
Mtime	X						
Ctime		X					
Size	X						
SHA1		X					
Originallocation		X					
Firststorageprotcol			X				X
Firststorageserver			X				X
Firststorageinplace					X		X
Firststorearrivaldatetime					X		X
Firststorageunpacked					X		X
Firststoragelastvalidated					X		X
Firststorageprocessed						X	
Secondstorageprotcol				X			X
Secondstorageserver				X			X
Secondstorageinplace					X		X
Secondstoragearrivaldatetime					X		X
Secondstorageunpacked					X		X
Secondstoragelastvalidated							X
Secondstorageprocessed						X	
Thirdstorageprotcol				X			X
Thirdstorageserver				X			X
Thirdstorageinplace					X		X
Thirdstoragearrivaldatetime					X		X
Thirdstorageunpacked					X		X
Thirdstoragelastvalidated							X
Thirdstorageprocessed						X	

Table 26 - inode table field updates by processing stage

6.8 Conclusions

In this chapter, we introduced and described FCluster and its component parts.

This design has been developed from the problem listing in the previous chapter using a Design methodology described in the appendix.

The form of the solution is a result of the belief that the best way to attain assurance is by means of a layer in the operating environment that cannot be subverted. We achieved this by designing a FUSE file system. FClusterfs deals with everything concerning the custodianship of data within FCluster. It is extensible, and so, even if this design is seen to be inadequate in certain measures, it can be amended to cater for future changes.

Early on, it became likely that the addition of this middleware would affect the performance of the system and so we developed both Jigsaw imaging and processing prioritisation to improve data flow through the system.

In summary the key features are:

- Acquisition of data by DEC;
- Prioritisation by criteria set by the investigator;
- The use of a relational database to store metadata about the evidence gathered;
- The provision of transparent access to the metadata database by using a FUSE file system;
- The dispersal of data across a distributed storage system;
- The resulting ability to conduct concurrent local processing on that data;
- Complete control over access to the data;
- The ability to separate access to the raw data from the access to the resulting analysis;
- The retention of access to the original raw data without having to work on abstracted data.

In the next chapter, we concern ourselves with evaluation.

7 Implementation and Prototype Test Rigs

In this chapter, we describe the implementation of FCluster ready for the evaluation stage. We state the objectives of the evaluation and the problems of selecting an appropriate platform between the choices of real hardware and virtual platforms.

7.1 Evaluation Objectives

From our hypothesis, on page 11, we have two objectives. Firstly to demonstrate the adequacy of the chain of evidence within FCluster and secondly to demonstrate the timeliness of this solution. Ideally we would simply build the target test environment of about 100 i7 processor based PCs. Unfortunately that is not possible within our research budget, so we must seek an alternative.

The most obvious alternatives are in virtualisation and cloud deployment. Both VMware Desktop and Amazon Web Services Elastic Cloud Compute (EC2) were used during the development of FCluster and were considered as candidates for the evaluation phase. Ubuntu 12.04 LTS desktop version was installed within VMWare Desktop version 10 running on an i7 desktop PC with Windows 7 64-bit, 24GB RAM, a 128GB SSD for the system and a 1TB hard disk drive for data. VMWare is capable of hosting up to about six virtualised hosts with this host, each as an 8-core processor with 2GB RAM and 10GB disk space. The same environment was created in Amazon EC2, which is effectively a virtual environment but operating in a remote location. EC2 instances run as virtualised machines on XEON hosts. Amazon EC2 does an advantage in that it has a variety of 'Instance Types' which enables identical software setups to run on variable hardware platforms. It seems that the c3.2xlarge is equivalent to an i7 and costs $0.42 per hour.

These work well when evaluating the user interface, functionality and interconnectedness of the software and hardware but it creates problems when used for timing test. One key objective in any testing scheme is to eliminate variable factors that would cause the results to change between what should be identical tests. Unfortunately, when we run software on operating systems such as Linux or Windows, there are a number of mechanisms that influence performance, and most of them are beyond the control of the operator. Task scheduling, disk caching, memory management and hard disk response times are just a few examples. Running these operating systems within a virtual environment, whether it is local or remote, makes this worse and even further beyond our control. Within a virtualisation environment, a control program, often called the hypervisor, takes charge of the resources allocated to each virtualised hosts. This automatic control prevents a stable evaluation environment being established.

To obtain absolute performance figures, trials should be run as near the real hardware as possible.

Amazon Web Services do offer a dedicated machine, the c3.8xlarge that is effectively the XEON host itself, at about $1.68 per hour. Unfortunately, these are effectively out of reach given the research budget and the time scale of evaluation.

7.2 VMware Desktop 10 for Chain of Evidence

As all development work was carried out within a virtualised host running Ubuntu Desktop version 12.04 with 4GB RAM, we selected the VMWare desktop 10 environment as suitable for proving the chain of evidence.

The development environment included

- One host for DHCP with MasqDNS;
- One host for development with gcc, ntfs-3g, MySQL, Gambas;
- Six workstations each hosting a MySQL server and ftp server.

The VMWare environment allows the creation of multiple virtual hard disks for imaging without the need for extra hardware although actual hardware can be attached via USB if required.

7.3 A LiveCD

Setting up a cluster can be a considerable administrative task; it is no surprise that most cluster solutions have an associated administrative package to enable the rapid deployment and updates of hosts.

When we assessed the practicalities of running forensics on a Globus Grid system in 2008, we started the research build facing the task of building a full Globus Grid system from source. Confirming Roussev’s observation in 2004 that “grid computing today is anything but easy to use or deploy”, we found building a Globus Grid system by hand was extremely difficult. Fortunately, we found that a LiveCD implementation, Instant Grid (Instant-Grid 2013), had been built. This included a DHCP server, full authentication and a number of scripts that allowed configuration of the primary host as it booted. This greatly reduced the task of creating the Grid.

For FCluster, we used a LiveCD creation tool, originally called Remastersys (Stafanov & Brijeski 2013) now called System Imager (Black Lab 2015), to make a single LiveCD/DVD delivery from our virtualised hosts. It differed from the Instant-Grid LiveCD in one key respect. Instant-Grid boots from just one single LiveCD in a master machine and all subsequent hosts used the master host as supplier for the operating system image that is accessed via a PXE boot that mounts an image from the original host’s CD. Throughout the functioning of the Instant-Grid cluster, all hosts needed constant access back to the primary host’s CD drive. This put great strain on the primary host and limited the size of the test cluster size to three or four hosts. Instead, we have chosen to deploy individual LiveCDs for each host on the cluster. This allows clusters of almost any practical size to be created very quickly. We have created temporary clusters of up to 60 host PCs with this technique. Additionally, remastersys provides a single click command to install the contents of the LiveCD onto the hard disk of the host PC if required. This allows us to create an exact copy of our LiveCD on the chosen hardware for performance testing.

As part of the boot procedure, the LiveCD operating system searches for a DHCP server. If it does not find one, it starts its own to provide DHCP services to the cluster. This will be the case if this is the first hosts in the cluster. If it does find one, it skips over the initiation of the DHCP server, using the primary host's DHCP instead.

As each LiveCD boots, it looks for hard disk drive 0, and if there is a suitable file-system, NTFS or EXT3/4, and enough space, it will create a temporary folder in which it will store its data. All other activities are actioned in a RAM Disk environment. As the boot progresses, it looks for an ftp server running on the host that provided the DHCP service. As the root crontab starts to run on each host, it connects to a specific external ftp server that holds a copy of the latest scripts and programs for the cluster and downloads a copy needed to run the cluster. It also draws information from a text file that defines the functionality that a host should fulfil. If no specific definition is found the host becomes a general-purpose storage/processing host for the cluster. The cluster host 'roles' are explained in detail in 6.2.3 on page 141. In this way, many identical LiveCDs can be created and do not need to be constantly updated with program amendments.

This deliverable means that setting up a test system is easy whether the tester runs the LiveCDs within a virtual environment or on an existing network with operating systems already installed.

7.4 Hardware for the timing test

Using the LiveCD, and so establishing constancy between implementations, we were able to boot on 'real' hardware. Our chosen system was:

- Lenovo Thinkpad i7 with 8GB RAM;
- Ubuntu 12.04 OS runs on a SATAIII - Crucial-M SSD;
- Input drive on USB;
- Image Output drive on USB;
- DEC output drive on USB.

7.5 Two Evaluation Environments

Subsequently we have two environments. Firstly, a virtualised environment that provides an adequate base for all assessments other than performance. For performance testing, we use the hard disk install facility from remastersys to enable us to create a replica system installed directly on hardware.

7.6 Sample Data

Jigsaw imaging is intended to be used when acquiring static data from large media, certainly bigger than 1TB, where conventional linear imaging takes many hours and so delays the investigation progress. It is most effective when acquisition requires the production of a full forensic image as well as selective data in DECs and where the large media is only partially filled with data with a high potential.

In selecting suitable test data one should try to select one that represents an accepted standard from within the domain. Woods et al. (2011) have created a corpus of media for research – Digital Corpora (Garfinkel 2014). Their focus is in creating realistic forensic corpora that is plausible and internally consistent. As they are attempting to prove data sets for educational and research purposes the samples they create and store are of limited size. There are currently about 150 images of hard disks with the largest being 80GB.

It seems unlikely that there will be a suitably sanitised large dataset available within the Corpus for some time but it is important that we should work on media with a representative real use. Subsequently, we tried to find our own candidate.

To do this we needed to obtain an overview of the contents of each media candidate to enable us to judge its profile and therefore its suitability for analysis. This was achieved by running a small meta-data collection program against each candidate disk.

A meta-data collection script was written as a Bash script and runs from an Ubuntu Linux LiveCD. In this way, we were able to operate the program without affecting the host PC. The script mounts drive 0 and reads the directory tree using a number of well-known Linux utilities such as “ls”, “fdisk”, “hwinfo”, ”ntfslabel”, “filefrag”, “file”, “md5sum” to elicit information about the target file system.

The following Table 27 was created in MySQL and was populated with data from the file-system under examination.

Field name	Type	Description
DiskID	char(32)	Hard Drive ID
NTFSSerial	char(32)	UID for File System
inode	bigint(20)	Inode record for the file
FileExt	varchar(50)	File extension
FileSize	bigint(20)	File Size in bytes
Fragments	mediumint(9)	No of Fragments in the file
Filename	varchar(1024)	Full path filename
LastAccess	char(15)	Last Access Data/Time
MD5Sum	char(32)	MD5 sum of File data
TypeDescription	varchar(300)	File type description from 'file' command
ID	bigint(20)	Unique record ID

Table 27 - File System Analysis Table Structure

We ran the analysis program on about 60 PCs but found that there was so much variation within such a small data set that there was little advantage in trying to form an idea of 'average' or 'typical' contents by any statistical means.

With the issues of privacy and of the availability of suitably sized media, it was decided to use a drive from a real machine, graciously provided by a friend. The script was run against this 256GB hard disk that had 325,372 files that occupied 168GB and was felt to be typical of many large hard disks found in notebook PCs. The content of this file-system was felt to be representative of a hard disk that has been subject to average use for about 18 months. We noted that the 256GB drive is now at the low end of drives sizes found in retail offerings, 1TB hard disks are more typical.

Even this is not large enough to demonstrate fully the advantages of Jigsaw imaging and prioritisation but it is three times larger than the largest in the Digital Corpora. Subsequently a larger disk of 500GB with 182,527 files that occupied 207GB was chosen as an alternative.

	GB	Files	GB Used	GB Free
Disk A - Phil	256	325,372	168	88
Disk B - Nick	500	182,527	207	293

Table 28 - Sample Disk Statistics

The two disks, listed in Table 28, do demonstrate the variation of disk contents with the former disk being twice the capacity but having 40% fewer files.

This 256GB drive is from a host running Windows 8 that is held within a single partition of 256GB with 168GB of data leaving 88GB free. The single partition contains the operating system, typical installed programs such as Microsoft Office and a collection of domestic videos of skiing and sailing holidays, JPG photographs of the same and the documents associated with Office. The PC has Outlook to collect email.

We found these disks to be typical of the set of 60 disks we examined.

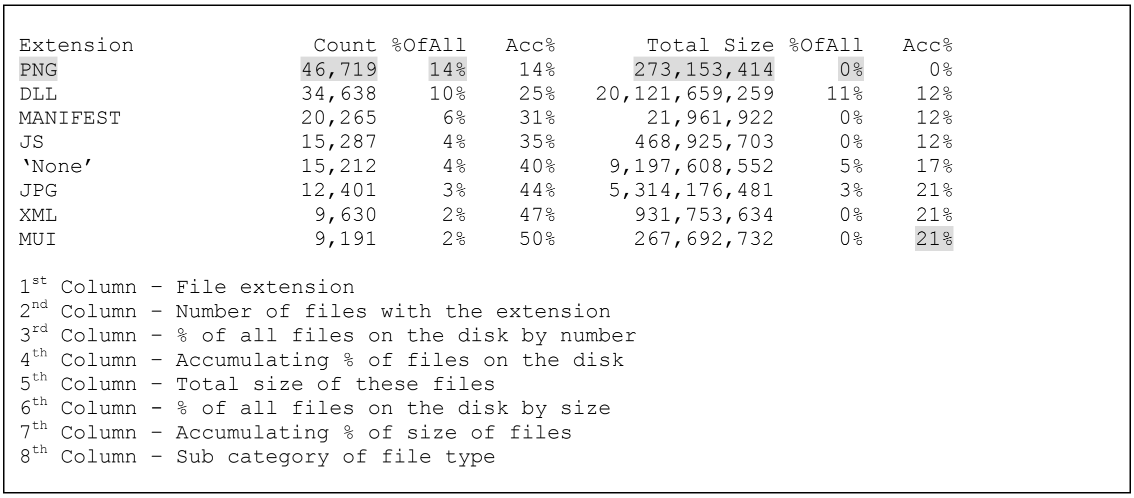
7.6.1 Statistics for Disk A

Appendix B has a table that contains the statistics of the files found on the 256 GB disk. In summary:

- There are 325,372 files in total;
- 97% of files are unfragmented;
- 2% of files have between 2 and 5 fragments;
- 29 files are over 1GB;
- 202,313 files (62%) are less than 10k.

7.6.1.1 By Count of Files

Figure 95, shows a listing of files, grouped and totalled by file extension and sorted into number of files.



```
Extension                Count %OfAll   Acc%       Total Size %OfAll   Acc%
PNG                     46,719    14%    14%      273,153,414     0%     0%
DLL                     34,638    10%    25%   20,121,659,259    11%    12%
MANIFEST                20,265     6%    31%       21,961,922     0%    12%
JS                      15,287     4%    35%      468,925,703     0%    12%
'None'                  15,212     4%    40%    9,197,608,552     5%    17%
JPG                     12,401     3%    44%    5,314,176,481     3%    21%
XML                      9,630     2%    47%      931,753,634     0%    21%
MUI                      9,191     2%    50%      267,692,732     0%    21%

1st Column – File extension
2nd Column – Number of files with the extension
3rd Column – % of all files on the disk by number
4th Column – Accumulating % of files on the disk
5th Column – Total size of these files
6th Column – % of all files on the disk by size
7th Column – Accumulating % of size of files
8th Column – Sub category of file type
```

Figure 95 - Top File Types by Number of Files

This shows that the most numerous file extension was a PNG, followed by DLL and then MANIFEST. There were 46,719 (the 2nd Column) PNGs (the 1st column) representing 14% (the 3rd column) of all files on the disk by number. These totalled 273,153,414 bytes (the 5th column), or 273MB, which did not even represent 1% (the 6th column) of the contents of the disk by size but 21% of all file space is occupied by files with one of just eight extensions.

It appears that 50% of all files on the disk have one of only seven file extensions or had none.

7.6.1.2 By Size of Data

The Listing in Figure 96 shows a listing of files, grouped by file extension and sorted into total size on the disk.

```
Extension               Count %OfAll   Acc%       Total Size %OfAll   Acc%
 MP4                      205     0%     0%   65,167,024,000    38%    38%
 DLL                   34,638    10%    10%   20,121,659,259    11%    50%
 SYS                    1,302     0%    11%   16,624,260,915     9%    60%
'none'                 15,212     4%    15%    9,197,608,552     5%    66%
 JPG                   12,401     3%    19%    5,314,176,481     3%    69%
 EXE                    4,330     1%    20%    3,986,747,297     2%    71%
 M2TS                       1     0%    20%    3,913,285,632     2%    73%
 CAB                      418     0%    21%    3,348,727,504     1%    75%

1st Column - File extension
2nd Column - Number of files with the extension
3th Column - % of all files on the disk by number
4th Column - Accumulating % of files on the disk
5th Column - Total size of these files
6th Column - % of all files on the disk by size
7th Column - Accumulating % of size of files
8th Column - Sub category of file type
```

Figure 96 - Top File Types by Total Size of Data

The file type occupying most space is MP4 (1st column) with 65GB data (5th column). There were just 205 (2nd column) of these files.

It shows that 75% (7th Column) of the file system was occupied by data from one of just seven file extensions or had no extension.

7.7 Conclusions

In preparation for the evaluation the setup and configuration of a test environment was found to be a series of compromises. In a world without limits, it would be best to build and test a full working version of the target system but this was not possible. Instead, we assessed the requirements of the specific evaluation and built several test environments that enabled the evaluation of specific features. Each of these environments provides stability in the aspects of the environment that could affect the specific evaluation results.

The same was true of the test data used. Reflecting back to chapter 3 and 3V, Velocity, Variety and Volume, we are affected by the same dilemma. Our conclusion was that it was not possible to use a form of ‘average’ data, as the notion of average makes no sense in the wide distribution we are required to investigate. Instead we chose what we considered might be expected to be presented as evidence.

In the next chapter, we will conduct the evaluation.

8 Results, Evaluation and Assessment

8.1 Introduction

In the chapter 6, we described FCluster by a detailing each of its component parts; Jigsaw Imaging, data prioritisation and FClusterfs. In chapter 7, we described the test rig and selection of data for our analysis testing. In this chapter, we evaluate FCluster and its components.

We first reflect back to the hypothesis and consider the best way to implement an evaluation. We find that there is a need to approach different aspects of the system with different types of evaluation. In section 8.2.1, we will consider different approaches to evaluation and conclude that a mixed approach, appropriate to the section of the system under consideration, is best. On the issue of 'timeliness', a quantitive approach is deemed best and on the issue of 'soundness' a qualitative approach is selected as best suited.

In 8.3, we provide an extension to the thoughts developed in 5.3, describing the problems, and 5.4, describing the solutions, to include pointers to the evaluations in this chapter.

In the following sections, we evaluate, in turn, Jigsaw Imaging, Processing Prioritisation, FClusterfs and then FCluster as a whole.

8.2 Evaluation Strategy

Referring back to the original hypothesis, in section 1.3.4, there are two characteristics that need evaluation:

- Timely Handling of processing;
- Maintaining Information Assurance primarily by maintaining chain of evidence.

As we have seen, FCluster comprises of several components. Each has its own characteristics and so may need to be evaluated in different ways.

8.2.1 Evaluation techniques

There are broadly, two types of evaluation available:

A quantitive one in which we time or measure the components and a qualitative approach in which we assess the capabilities of the components and consider their fitness for purpose or conformance to a standard. Quantitative evaluation generally attracts greater merit than qualitative evaluation.

8.2.1.1 Quantitative

In a quantitive evaluation, we attempt to obtain a numeric measure of the subject in question. This enables a comparison within the subject, for example to measure change, or to allow comparison between the subject and some external system. Roussev (2004) observed the difficulty in comparing his prototype of DELV with existing systems like FTK. Having created a target 'typical' image, Roussev processed it using FTK in just under 2 hours but observes:

> *"None of the measurements for our prototype are directly comparable to these numbers, because FTK is performing a lot of initial pre-processing of the (forensic) image and we only have a general idea of the implementation "*

The speed and processing performance of FCluster is clearly capable of evaluation in a quantitative manner. However, this is not the real focus of FCluster. Speed is important but, at least initially, only as a measure of the penalty of the increased management suffered by additional processing that does no more than increasing assurance. We have no doubt that if working in isolation, and all other things being equal, ten computers can do ten times the work of one computer. What is of concern is the effect the act of load balancing, distribution of data, initiation and completion of tasks has on the performance.

There are elements of this research that can be subjected to quantitative analysis. For example, we could complete a linear image with, perhaps dd, dcfldd or d3dd and then a comparable run with Jigsaw imaging. It is to be expected that Jigsaw would be slower but just how much slower will it be? However, we must ask, is this comparison of any use because it runs into the same problem as previously identified by Roussev in that the outputs of dd, dcfldd and d3dd are not the same as Jigsaw imaging? If this was done, then a more realistic proposal might be to run dd to get an image, then run AFF to create the DECs and compare this overall with running one pass of Jigsaw imaging. However, this does not take account of the feature of Jigsaw imaging that delivers DECs for processing before the whole imaging process has finished.

It seems likely that an attempt to compare FCluster and Jigsaw imaging with existing techniques will suffer because of the same reasons identified by Roussev.

8.2.1.2 Qualitative

Although assurance can be measured on a scale of, for example, 1 to 10, it is really a question of a qualitative assessment. We could equally use a scale of A to J. A qualitative evaluation is often based upon an opinion of an individual's, or group's, contentment with the attainment of a criterion, for example a standard of operation. Specifically, in the case of the standards we considered relevant to this work, it is usually the real world case that an external auditor will assess the situation presented to them against a criteria, such as an ISO standard or the APCO guidelines.

When implementing an ISO 27001 Information Assurance Security Management System it is considered best practice to start by conducting a 'Gap Analysis' to establish the state of the security management system before, during and after implementation. This allows a baseline to be established and clarifies areas that need attention to achieve the certification to the standard.

In more detail shown in Figure 97 – Gap Analysis Evaluation Process, the standard is assessed, a business plan is written describing the intended end result, a formal 'Statement of Applicability' (SoA) is written which describes how the standard will be implemented to fulfil the business plan. A Gap Analysis is conducted in which the current state of the business is compared to the vision for the future. An implementation plan is then created to guide the implementation. Throughout the implementation, the Gap Analysis is revisited to assess progress and priorities in future work. Finally, the newly implemented management structure is assessed by external auditors and, if appropriate, certification is sought.

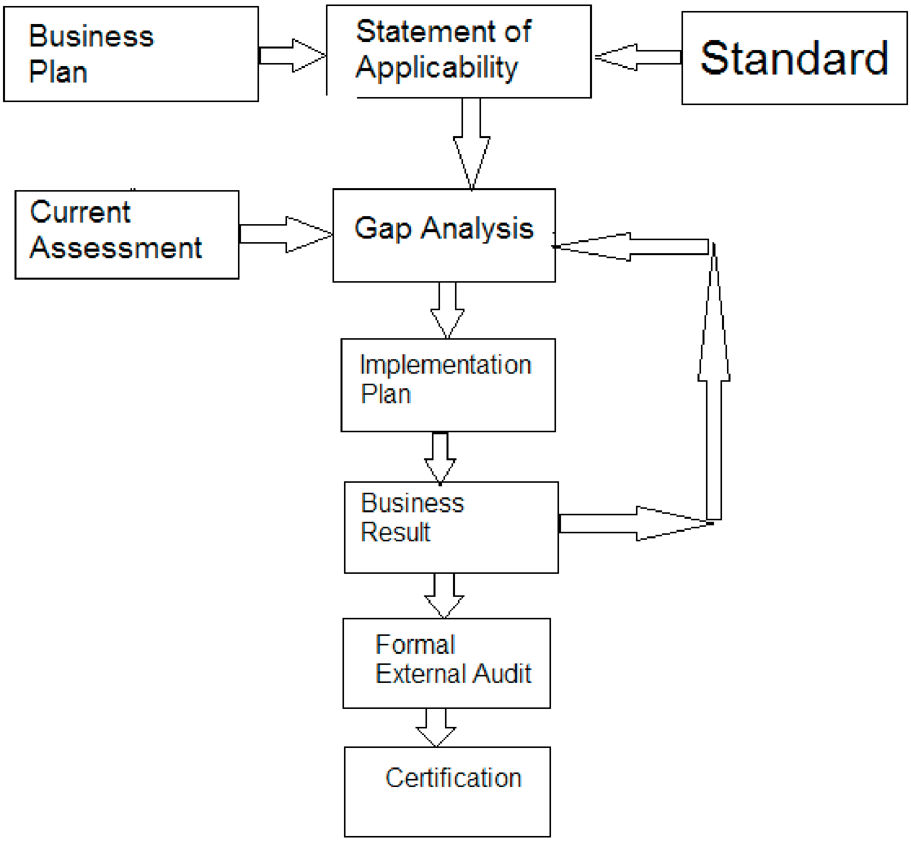


Figure 97 – Gap Analysis Evaluation Process

A qualitative assessment of the assurance provided by FCluster and FClusterfs against a collection of relevant assurance standards would be more productive as that is what the design was intended to address. Table A, in the Appendixes, continues the previous columns that listed the problems and solutions to provide a corresponding pointer to the evaluation in this chapter.

8.2.2 Evaluation Applicability

As we saw in chapter 6, FCluster consists of a number of components. Some are best evaluated in a quantifiable manner; some are best assessed by qualitative means. Table 29 shows the components parts and the applicability of qualitative and quantitive evaluation. The table then identifies against which criteria they are best assessed.

System/ Feature	Evaluation Criteria	Evaluation	CIA	ACPOv5	NIST Data Acquisition Test	ISO 27037 2012	ISO 17025 2005	ISO 14721 2012	Daniel Ayers	Timed
Jigsaw Imaging	There is no doubt that the extra processing in Jigsaw imaging, above that of simple linear imaging, will make it slower.	Quantitative								✓
	Is Jigsaw capable of a complete image?	Qualitative	✓		✓					
DEC	The integrity of the resulting DECs must be maintained by a robust structure for the acquired data in within the DECs.	Qualitative	✓	✓		✓	✓	✓		
Prioritisation	Time the setup of the prioritisation database and sort into order. By demonstrating the improvement in delivery of data for processing.	Quantitative								✓
FClusterfs and FCluster	Robustness and failure to maintain Chain of Evidence. By assessing the facilities of FClusterfs against appropriate standards and best practices.	Qualitative	✓	✓		✓		✓	✓	
	Management overhead affecting processing speed	Quantitative								✓

Table 29 - Evaluation Strategy – Overview

8.3 Jigsaw Imaging

8.3.1 The Evaluation Criteria

In our evaluation, we restrict ourselves to two points, speed and integrity. Jigsaw imaging is the creation point for the DECs that will be drawn into the FCluster system. The integrity of the evidence must be established and recorded at the point of acquisition. Ideally, this should be done with as little possible effect on the delivery time of analysable data.

8.3.2 Existing Evaluation Standards

Disk, or media, imaging is one of the areas covered by the National Institute of Science and Technology's (NIST) testing regime for Computer forensics Tool Testing (CFTT). Currently just over 30 products have been tested against the current Acquisition Tool Specification - Draft 1 of version 4.0 (National Institute of Science & Technology 2005) and Digital Data Acquisition Tool Test Assertions and Test Plan - Draft 1 of Version 1.0 (National Institute of Science & Technology 2005). The full list has 34 required and optional features. Our evaluation of a prototype version and so cannot be expected to fulfil all these requirements.

The NIST tool test assertions all focus on thoroughness, integrity, recovery from error and operator feedback information.

There are two optional tests within the NIST criteria DA-OA-14 where a clone is accurately written to the same address on the clone, which is true. In addition, DA-OA-22, that the tool creates accurate block hashing, would apply, which is also true.

None of the NIST tests covers the time taken to acquire an image. Subsequently, we choose to setup a practical test and time the result.

Our assessment will be between 'conventional imaging' in the form of a complete media image using *dcfldd* and a complete image using Jigsaw but with the creation of a sensible number of DECs during the process.

8.3.3 The Testing Method

We will conduct two tests. On timeliness, it is a straightforward timing to achieve the same copy with Jigsaw and dcfldd. On integrity, we will then use a small bash script that calls the Linux utilities *diff*, *du* and *df* to check the imaging and unpack the bagged data to check the encryption and encoding in the DEC.

Timing comparisons must accept the limitations of the test environment. Obviously, the process can be improved by adding Solid State Devices or faster interfaces but these do not test Jigsaw imaging, they test the environment. Timing is more a question of relative results comparing the proposal against the previously accepted solution and observing any increase or decrease in performance.

8.3.4 Prediction on timing

Assuming the linear read speed remains constant, the expected speed degradation with Jigsaw imaging comes from two actions. Firstly, random access seek times when moving between the starts of data runs, e.g. at the start of reading a file and similar seek actions when reading a file that is fragmented. Secondly, the additional processing needed to select the potentially high value files, calculate the cryptographic hashes and bag the evidence into the DECs.

With regards the random seeks, we observe that it is now common for hard disks in typical use in desktop PCs or notebooks to contain, perhaps, 250,000 files, which means 250,000 seeks to the starts of these files. A 2TB NTFS file system consists of 500,000,000 4kb clusters. The data transfer rate from a hard disk is a complex matter and consists of several components. Manufacturers often quote the rate at which data can be retrieved as a constant stream from the device. This is typically 80-120 MB/s. The most important effect that degrades this access rate is that of the head seeking across the disk, which is typically 9ms = .009 seconds.

We suggest that even if the 250,000 random seeks to the beginning of the files are 1000 times slower than the sequential reads the overall impact is to add less than 5% to the overall time.

When linear imaging, if a 2TB disk has 4,000,000,000 x 512 byte sectors and takes 7 hours to image then each sector must take 0.00005 seconds to access and read.

When Jigsaw imaging if most sectors take 0.00005 seconds but the first sector of each file takes .009 seconds, nearly 200 times longer, to access and read this means that Jigsaw imaging should take just 250,000 x .009 seconds longer to complete the same task. Therefore, we expect Jigsaw to be about 37 minutes longer on a 7-hour task, which is 9% more.

Additionally, regarding seek times when reading a fragmented file, we make two observations. We find the new huge disks are often only sparsely filled, and so when files are written they do not need to weave between used clusters. In addition when they are filled it is often with write once, read after, files like video. In addition, Windows 8 now runs defragmentation as a scheduled event as a standard configuration item.

Solid State drives have zero seek times and so Jigsaw should see no speed penalty at all.

Subsequently, we do not feel this overhead is an issue.

Secondly, on the matter of processing overhead, we believe that running this on an i7 processor will provide ample power to complete these tasks without any significant time penalty.

Jigsaw imaging's main claim is to deliver actionable data very quickly and this is beyond dispute. From initiating imaging, it is likely that the first evidence could start to be delivered as single DECs within less than a minute. Subsequent DECs containing high valued data then roll off as fast as it can be written to the collecting device.

8.3.5 Actual results on Timing

We used one of our sample disks, which is 500GB, with 182,527 files totalling 207GB with 293GB free, as it is the largest and so more of a challenge.

We ran "*dcfldd if=/dev/sdl of=/dev/sdm*" and it took 513 minutes on our machine. That is 16.2 MB/s.

We ran Jigsaw with a regular expression of

"/.PF$|/.PDF$|/.DOC$|/.XLS$|/.TXT$|/.JPG$|/.C$|/.SCR$"

The run was with the regular expression above and was used to create DECs for all the files *.PF, *.PDF, *.DOC, *.XLS, *.TXT, *.JPG, *.C, and *.SCR, these numbered 6720 and totalled 13GB. They were finished after 40 minutes.

The remaining 175,807 files were finished after a further 260 minutes. The remaining unallocated clusters took an additional 500 minutes to copy. Therefore, Jigsaw imaging took 12:40 to complete, roughly 33% longer.

Action	Volume GB	Minutes to complete	Rate MB/s	Speed + is slower
Dcfldd	500	513	16.2	Datum
Copying and bagging 6720 files	13	40	5.4	+67%
Copying 175,807 files	194	172	12.4	+23%
Copying Unallocated space	293	460	10.6	+35%
Total for Jigsaw	500	672	10.9	+33%

Table 30 - Jigsaw Imaging test results

The full results, in Table 30, show that the first stage, copying and bagging, took 67% more. This was greater than expected and is probably because storing the bags requires that a file has to be created, opened, written to and closed within an EXT4 file system.

If the entire source drive contained data then we could expect the overhead in the first stage to be continued throughout the whole process extending the entire process to 856 minutes.

The 35% speed reduction while copying the clusters that constitute the unallocated space was a surprise as the task is so similar to the process in dcfldd. It may just be down to the efficiency of the code.

One significant observation was that while imaging by either linear or Jigsaw, the processor was hardly working, staying at about 25% capacity. Because Jigsaw collects data in files, it is available for on the fly processing within the remaining 75% capacity of the processor.

8.3.6 On Integrity

The tests run with various Linux utilities proved that the imaging and bagging was reliable.

On the issues of resilience to interruption and recovery from error, neither of these issues were addressed in the prototype code. The obvious technique is to create and maintain an array to parallel the NTFS $Bitmap file used to locate data on the original evidence media. In this sense, Jigsaw imaging has the same efficacy and problems as its peers as it could be viewed as a linear imaging applied in bursts controlled by the directory file $MFT.

The same can be said about achieving thoroughness. The $Bitmap file provides a mapping of all the clusters within the partition, when this is complete, so all the cluster have been copied.

8.3.7 Assessment of Gains and Losses

During a linear imaging process, it is possible create a cryptographic hash of the source media stream and so enable the destination to be verified after it has been written. This does require the destination to be re-read in its entirety as a stream and so consequently, it is time consuming but the ability to verify the whole copy is highly valued as a means of assuring integrity. This is common practice to write this process into an acquisition procedure.

During Jigsaw imaging's non-linear process, we lose the cryptographic hash of the entire image as a single entity. This would be true of any non-linear imaging, for example, AFF, the EnCase Logical Evidence File format or DEC as we reviewed in section 4.6. Similarly, we do gain the ability to record the cryptographic hash of each individual file that is not available in linear imaging, as it does not read files.

With the ever increasing use of cloud storage we should ask for how long will we have the ability to create whole media hashes anyway? We gain the advantage of cryptographically hashing each individual file. As Jigsaw imaging reads files we gain the advantage that we can check teach file against a hash database to enable the DEFR to be alerted to 'contraband'. Current linear systems can only read individual clusters, create a hash and compare it with a database of contraband clusters hashes.

8.3.8 Conclusions

Jigsaw imaging has a place in an arsenal of acquisition techniques. It clearly scores well when used in large media that is only partially filled. Its primary advantage is that it allows access to the data selected by the investigator within a few seconds rather than have them wait until the imaging and verification is complete without the need to start again on imaging.

8.4 Prioritisation

8.4.1 Introduction

Figure 98 - Accumulated Evidence Score is provided as an illustration of our objective. The x-scale is linear from 0 bytes to 168GB, representing volume of data read and processed. The y-axis is the accumulated score of the potential to provide evidence.

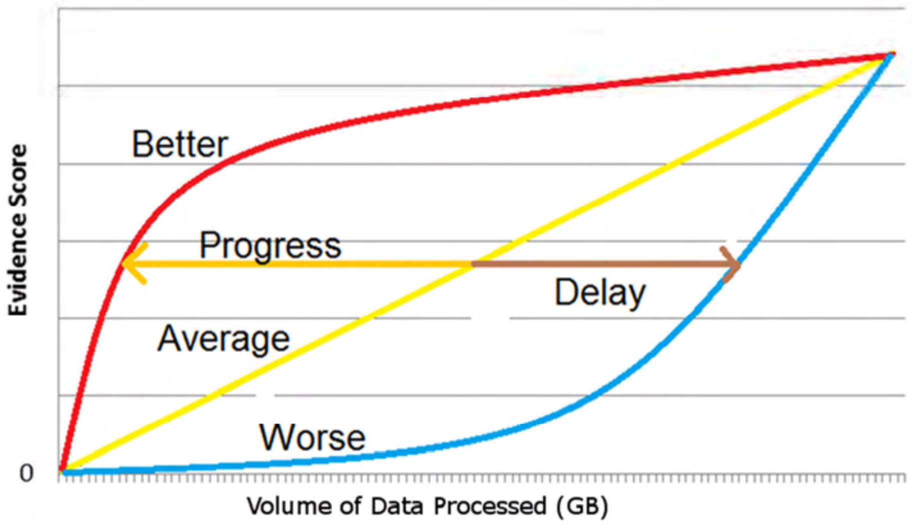


Figure 98 - Accumulated Evidence Score

If the files all had the same potential evidence score per unit of data then we would expect to see the yellow line. As data is read, evidence potential increases in proportion.

This would, for course, imply that a 5MB DOC file would be seen by an examiner to potentially hold the same chance as containing evidence as 5 JPG files of 1MB. This is the case without any prioritisation. Clearly, this is not a realistic expectation of the real world and is not suggested. The actual potential value of each type of data and its weighting based on size will vary from case to case.

Our prioritisation technique, described section 6.6, allows us to allocate score points to each of nine attributes. If we sort files based on these figures we will either draw files to the left, in red, representing more prompt processing or to the right, in blue, for less prompt processing.

8.4.2 Analysis Summary

Figure 99, Figure 100 and Figure 101 show our primary analysis.

8.4.2.1 By Evidence Score Card

The first table shows the file types, in order of Bayesian score allocated to the file type within the experiment. This list, shown in part in Figure 99 – Files Types sorted by User Allocated Bayesian Score, is sorted into the numerical order by the value allocated by 'the user' in parameter #4 of Table 14 - Evidence Value Score Parameters and listed. With 17 points awarded to AVIs, JPGs, MP4s and 16 points to DOCs, XLSs, DOCXs, F4Vs and so on.

Ext	Count	Total Size	Category	Value	Description
AVI	28	457,528,844	Video	17	Audio Video Interleave File
JPG	12,401	5,314,176,481	Raster	17	JPEG Image
MP4	205	65,167,024,000	Video	17	MPEG-4 Video File
DOC	394	164,337,404	MOffice	16	Word/WordPad Document
XLS	210	17,163,288	MOffice	16	Excel Spreadsheet
DOCX	1,001	266,899,785	MOffice	16	Microsoft Word Open XML Document
F4V	5	571,275	Video	16	Flash MP4 Video File
XLSX	50	1,447,020	MOffice	16	Microsoft Excel Open XML Spreadshe
PNG	46,719	273,153,414	Raster	15	Portable Network Graphic
FSD	1,575	841,482,240	MOffice	15	MSOffice File Cache
HTML	7,398	78,132,803	Web	14	Hypertext Markup Language File
HTM	4,862	93,322,059	Web	14	Hypertext Markup Language File
VB	2,943	6,249,082	Develop	14	Visual Basic Code
XLSB	2	1,647,139	MOffice	14	Excel Binary Spreadsheet

Figure 99 – Files Types sorted by User Allocated Bayesian Score

8.4.2.2 Post Score Allocation reports

The score factors from all the parameters in Table 14 - Evidence Value Score Parameters, on page 190, was run against all the files on the file-system including those shown in Figure 99 – Files Types sorted by User Allocated Bayesian Score, it is no surprise that the results, in Figure 100, show the top scoring individual files are all JPGs.

```
Score       Size LastAccess  K  S  C  E  D  L Ext     Dir               Filename
 882  1,636,491 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_094739.jpg
 882  1,950,514 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_094745.jpg
 882  1,708,496 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_094749.jpg
 882  1,620,133 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_094752.jpg
 882  1,908,146 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_102022.jpg
 882  1,708,212 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_102033.jpg
 882  1,667,173 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_102049.jpg
 882  1,243,931 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_104059_1.jpg
 882  1,234,623 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_104100_10.jpg
 882  1,217,424 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_104100_11.jpg
 882  1,207,309 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_104100_12.jpg
 882  1,170,692 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_104100_13.jpg
 882  1,148,884 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_104100_14.jpg
 882  1,294,962 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_104100_2.jpg
 882  1,261,563 2014-03-01  0  9  8  9  4  9 JPG     /Users/SkyDrive/  Photos 20131031_104100_3.jpg
```

Figure 100 - Top Scoring Files

In Figure 100, columns K, S, C, E, D, L contain the scores from the attributes described in Figure 20. These are explained in Table 31

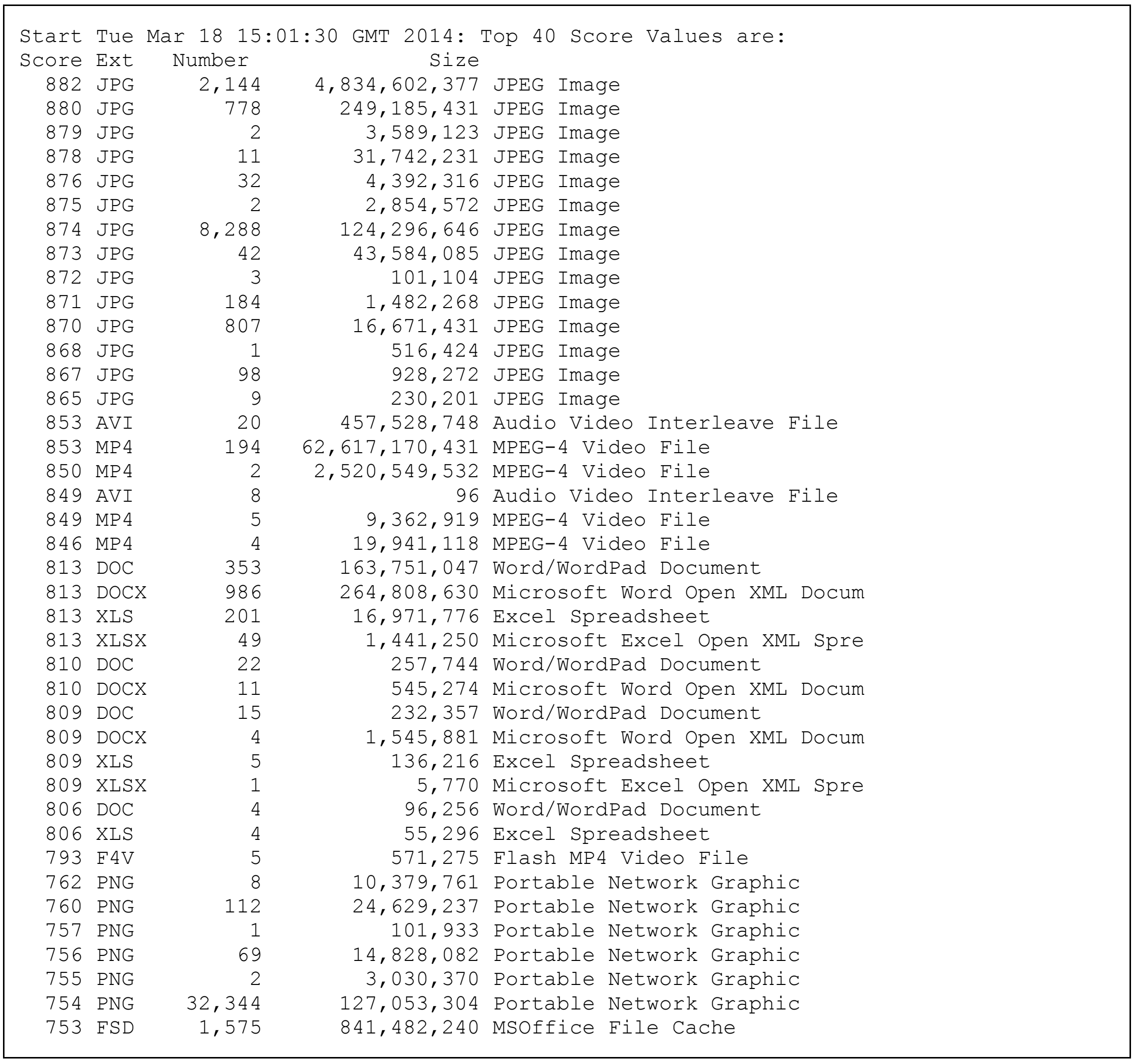
	Description	Value	Factor	Total Score
K	#7 – Known File Fingerprint	0	1000	0

S	#3 – Size of the file	9	30	270
C	#5 – File type category	8	25	200
E	#4 – File Extension	9	30	270
D	#6 – Last Modified Date	4	13	52
L	#9 – Processing Rate	9	10	90
				882

Table 31 - Allocation of Bayesian Values to Figure 95

Grouping files by similar score and file type, shown in Figure 101 - The Top score File Types, from 882 to 753 points

we see that there are 2,144 files with the highest score of 882 and they are all JPGs. The next 13 highest scoring groups are also JPGs. The variation in scores is because of date variations between the JPG files. We know, from Figure 95, that there are 12,401 JPGs on the media. However, there are only 12,371 listed before the next file groups that include AVIs, MP4s, DOCs and XLSs. This reflects the influence of other parameters, most likely date variations.



```
Start Tue Mar 18 15:01:30 GMT 2014: Top 40 Score Values are:
Score Ext   Number              Size
  882 JPG     2,144    4,834,602,377 JPEG Image
  880 JPG       778      249,185,431 JPEG Image
  879 JPG         2        3,589,123 JPEG Image
  878 JPG        11       31,742,231 JPEG Image
  876 JPG        32        4,392,316 JPEG Image
  875 JPG         2        2,854,572 JPEG Image
  874 JPG     8,288      124,296,646 JPEG Image
  873 JPG        42       43,584,085 JPEG Image
  872 JPG         3          101,104 JPEG Image
  871 JPG       184        1,482,268 JPEG Image
  870 JPG       807       16,671,431 JPEG Image
  868 JPG         1          516,424 JPEG Image
  867 JPG        98          928,272 JPEG Image
  865 JPG         9          230,201 JPEG Image
  853 AVI        20      457,528,748 Audio Video Interleave File
  853 MP4       194   62,617,170,431 MPEG-4 Video File
  850 MP4         2    2,520,549,532 MPEG-4 Video File
  849 AVI         8               96 Audio Video Interleave File
  849 MP4         5        9,362,919 MPEG-4 Video File
  846 MP4         4       19,941,118 MPEG-4 Video File
  813 DOC       353      163,751,047 Word/WordPad Document
  813 DOCX      986      264,808,630 Microsoft Word Open XML Docum
  813 XLS       201       16,971,776 Excel Spreadsheet
  813 XLSX       49        1,441,250 Microsoft Excel Open XML Spre
  810 DOC        22          257,744 Word/WordPad Document
  810 DOCX       11          545,274 Microsoft Word Open XML Docum
  809 DOC        15          232,357 Word/WordPad Document
  809 DOCX        4        1,545,881 Microsoft Word Open XML Docum
  809 XLS         5          136,216 Excel Spreadsheet
  809 XLSX        1            5,770 Microsoft Excel Open XML Spre
  806 DOC         4           96,256 Word/WordPad Document
  806 XLS         4           55,296 Excel Spreadsheet
  793 F4V         5          571,275 Flash MP4 Video File
  762 PNG         8       10,379,761 Portable Network Graphic
  760 PNG       112       24,629,237 Portable Network Graphic
  757 PNG         1          101,933 Portable Network Graphic
  756 PNG        69       14,828,082 Portable Network Graphic
  755 PNG         2        3,030,370 Portable Network Graphic
  754 PNG    32,344      127,053,304 Portable Network Graphic
  753 FSD     1,575      841,482,240 MSOffice File Cache
```

Figure 101 - The Top score File Types, from 882 to 753 points

8.4.3 "Evidence Points" Score Against "Time and Data Read"

The effectiveness of the prioritisation technique process is best illustrated by focusing on just one file type.

8.4.3.1 Selecting a high value specific file type – Only JPG

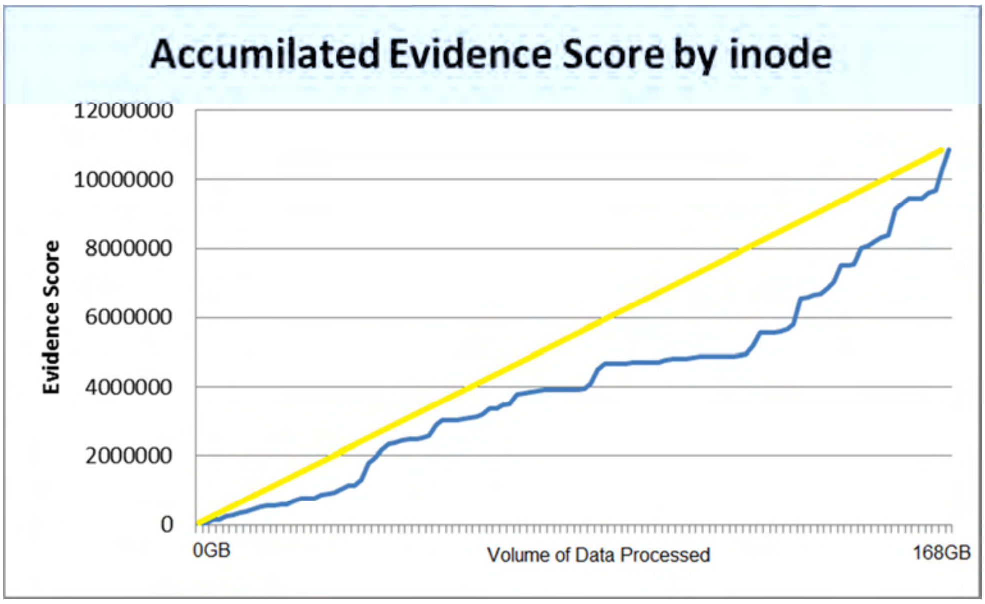


Figure 102 - Only JPGs by inode

In Figure 102 - Only JPGs by inode, we see the evidential potential points score attainment rate if only JPG files are selected as the files are scanned in inode order. Of the 130,000,000 'Potential Evidence Points' available on the whole file-system about 11,000,000 come from JPG files. Although there is a general trend to attain more points as the files are processed, it can be seen that in lower inode places evidence potential attainment is less rapid but towards the higher inode places it increases. This is expected as files that occupy lower inode places are usually occupied by files associated with Operating System installation and there are few JPGs to be found. Towards the end, in the higher inode places, the rate of potential evidence attainment is greatest. This is most likely because the user has added more JPGs to the file-system and these are added in the higher inode places. This sequence aligns to the "Worse" profile in Figure 98.

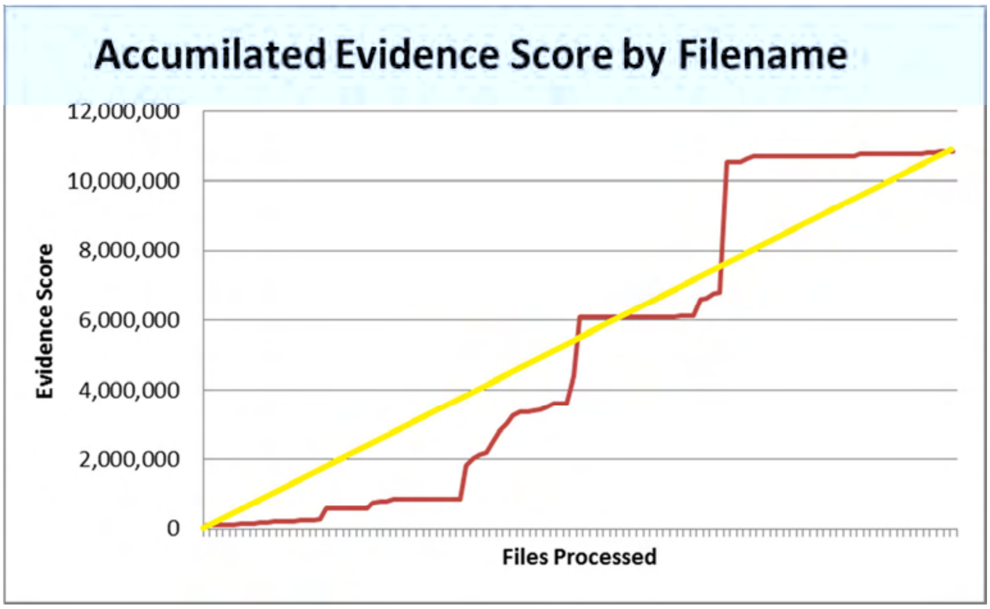


Figure 103 - Only JPGs by Filename

When the scan is executed in filename order, Figure 103 - Only JPGs by Filename, we see noticable steps of increase. It is likely that these are a result of the organisation of JPGs into directories that represent the subject or event being photographed.

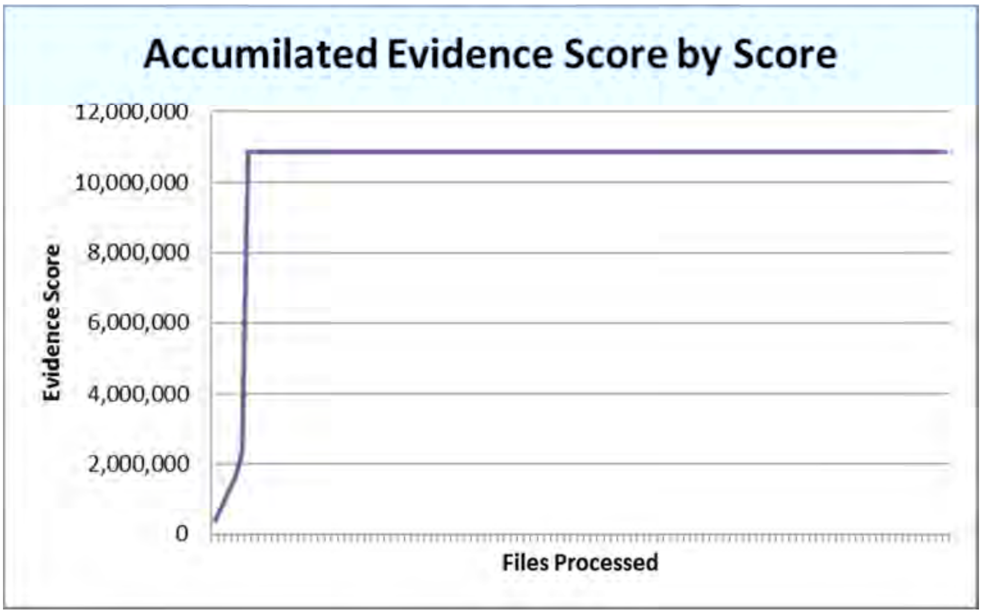


Figure 104 - Only JPGs by Potential Evidence Score

Figure 104 - Only JPGs by Potential Evidence Score, shows the potential evidence attain when sorted using the user allocated scoring system. 100% of the 11,000,000 associated with JPGs is achieved very rapidly. The first stage is not vertical because other file type, notably AVIs, were considered more valuable see Figure 99 – Files Types sorted by User Allocated Bayesian Score.

8.4.3.2 Selecting a low value specific file type – Only DLL

The opposite affect can be seen if we select a file type that we know has been set to a low potential evidential value. We can see, in the 3 graphs of Figure 105 - Selecting Only DLLs, how the use of the Potential Evidence Score delays any processing of DLL files until nearly half way through the processing cycle.

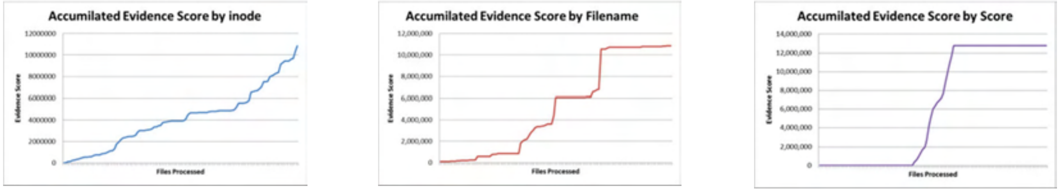


Figure 105 - Selecting Only DLLs

The first two graphs display the same sort of profile as we saw when we selected only JPGs but the final graph shows the same sort of rapid climb but it is delayed until much later in the overall process.

8.4.4 General Results

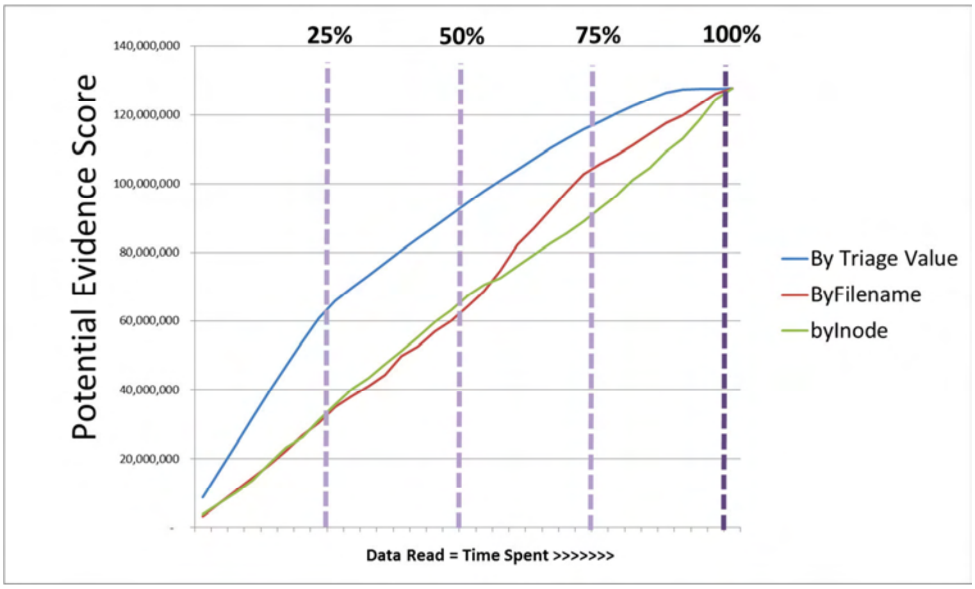


Figure 106 - Prioritising Processing

In 7.6, we stated that our test data set was a 256GB solid state drive from a well-used Notebook running windows 8. The total ‘Evidence Potential’ for the files contained in this file-system was about 130,000,000.

It can be seen, in Figure 106 - Prioritising Processing, that when processing the data by both Filename and Inode sequence we obtain a nearly linear increase in potential evidence value over time spent reading the data by volume.

However when we processed data by sorting our ‘Potential Evidence’ score the blue line align to the ‘Better’ profile in Figure 98.

- When processing by inode or filename sequence:
  - After 25% of the time we attain approximately 35,000,000 score (25%), of the potential evidence value;
  - After 50% of the time we attain approximately 60,000,000 score (50%) of the potential evidence value;
  - After 75% of the time we attain approximately 100,000,000 (75%) of the potential evidence value.
- When we use the Potential Evidence value:
  - After 25% of the time we attain the 60,000,000 score, nearly 50%, of the potential evidence;
  - After 50% of the time we attain the 95,000,000 score (75%) of the potential evidence value;
  - After 75% of the time we have processed 115,000,000 (90%) of the potential evidence value.

If the total evidence was, instead of 16GB perhaps 4TB, it might take 70 hours to process it all at 2MB/s on a 12 core i7 based PC. Processing by ‘Potential Evidence Score’ would reduce the time to reach 50% of the evidence by 25% overall and is, in effect a doubling of processing effectiveness.

This effect is even more dramatic when we assess the effect on specific file types.

8.4.5 Discussion

This technique enables prioritisation of processing data by allocating a hypothetical ‘Potential Evidential Value’ to a number of characteristics of file meta-data. This is more than just selecting certain file types for processing first. It allows more sophisticated prioritisation to control processing. In our examples, with a 256 GB disk we know to contain 168 GB of data, we improved the attainment rate by about 25%; see Figure 106 - Prioritising Processing. If this was applied to a 4 TB drive full of data, which would probably complete processing in about 4 days at 12 MB/s, we could reduce the time in which the data was processed by as much as 90% for specific file types. This is shown in Figure 104 - Only JPGs by Potential Evidence Score, and by 25% overall, Figure 106 - Prioritising Processing.

Even within groups consisting of large numbers of similar files, for example there are 12,401 JPGs in our example in Figure 95, the higher priority files will be pushed to the front of this sub-queue.

When the x-axis equates to processing the data on a 4TB drive these time reductions can be significant. If we consider our rule-of-thumb to process about 2MB/s on a single core in an i7-

based host, 4 TB of data will take about 4 days. This prioritisation technique could reduce results availability by a day or two.

There remain two obvious questions that remain unanswered in this test.

There is a question about whether to processes one large file rather than lots of small files. This is a strategic decision for the investigator who sets the prioritising parameters. Which is more likely to yield results sooner; Processing 10,000 x 1MB JPGs or 1,000 x 10MB JPGs? Both would likely take the same time but the former would 'cover more ground', in which case we might choose the adversely weight JPG files when they exceed about 5 MB.

The second question is in the area of job queuing and is associated with the previous question. Within a single PC there is little scope for distributing tasks so that one large task, for example still thumbnail generation of video frames from a large video file while concurrently processing many smaller JPG files. In a distributed processing environment, this becomes a real possibility.

What this prioritising technique does lack is the ability to choose and implement a strategy based upon the properties of the files to be considered. This could be in the form of a decision tree that allocated values dynamically during acquisition thus moving items up and down the priority list based on the amended values.

8.5 Digital Evidence Container Design

We explained in section 6.5 that a real implementation would most likely use a well-established DEC format, perhaps AFF. We found that using this in our prototype development became too complicated and so we developed a basic design for design trials.

The design, shown in Figure 88 and Figure 89 was explained in section 6.5. Its key claim to integrity is that each DEC holds a cryptographic hash of each cluster, a cryptographic hash for the entire file and that the header should contain sufficient data to enable its exact placement as part of a reconstruction of the original after imaging.

As a prototype, the design is incomplete and one obvious addition is that there is no record of the Master Boot Record details within the Master DEC data.

Additionally, our DEC naming scheme is flawed. The naming convention, shown on page 183, uses the file system serial number and the SHA-1 cryptographic hash of the contents. We found that host machines sometimes have their hard disks cloned from a master. This is often the case within large organisations. In addition, using the cryptographic hash of the contents means that these could be collisions within the naming of duplicate files. We note that AFF uses a unique, random ID string to name each container within its namespace.

8.6 FCluster and FClusterfs

8.6.1 Confidentiality, Integrity and Availability

In 3.4, we identified Confidentiality, Integrity and Availability as three founding principles of information security. Here we assess FCluster and FClusterfs against these principles.

8.6.1.1 Confidentiality

Confidentiality Assurance is present throughout FCluster. Jigsaw imaging reads from the original source and creates Digital Evidence Containers as one of two outputs. Our simplified DEC inherited a key feature from most of the existing forensic image and DEC design, that of encryption.

The key to be used in the encryption is generated within FCluster and then uses a PKI architecture to encrypt this using the public key of the designated imaging device together with the private key of FCluster. This is transferred to the imaging device where it is unpacked and will be used to encrypt all DECs.

Data stored within the DEC is encrypted using a cryptographically strong algorithm, AES-256. This data is then Uuencoded to reduce data transfer problems that can sometimes occur with non-ASCII data.

The DECs are transferred to FCluster be ingested. The identity of the DECs is verified and they are allocated and transferred to their allotted storage location while still in an encrypted form.

It is only when the DEC is received within the storage location that it is unpacked. Using the key, stored within the FClusterfs SQL database, the data is unpacked and stored within the file system using the FClusterfs FUSE file system. FClusterfs inherits the on-the-fly encryption technique used in eCryptfs, described in 4.5.4, with AES-256 and a locally generated encryption key to encrypt the data before it is stored on the media.

As the data is needed for processing, it is read from the media and decrypted in user memory space.

All data transfer is over, typically, SSH encrypted communications protocols.

Access to data within the FCluster file-system is controlled by entries in the SQL database behind FClusterfs. This can offer fine-grained access control to data.

8.6.1.2 Integrity

As the original data is read from the source media, a cryptographic hash is calculated for each cluster read and for each file read. A file is of course, a collection of a sequence of one or more clusters, so both can be calculated at the same time with one read of the data. This is recorded in the header section of the DEC created to hold the data. This meta-data is transported within the DEC, into FCluster. Finally, it is stored in the meta-data table of the SQL database used by FClusterfs.

Periodically, it is used by the verification daemon to check the integrity of the data as stored on the storage media.

There is a failing in this approach caused by the non-linear jigsaw imaging technique. Because cryptographic hashes are generated from a stream of data, it is no longer possible to create a cryptographic hash of the whole source media in Jigsaw imaging, as has been the practice for several years. ACPO 2.2.5, our section 8.6.2.2.

Selective and reasonable seizure is currently the subject of much discussion. Because of the difficulty of dividing the data on storage media because it has been considered indivisible, it has become common practice to image the whole media; as a result, a cryptographic hash of the whole media can be created. The 4th amendment of the Constitution of the United State (Legal Information institute 2014) contains reference to "[t]he right of the people to be secure [....] against unreasonable searches and seizures," and this being used to hamper collection of evidence in the form of an image. When this is set against the trend to adopt cloud storage systems and the popularity of 'closed' devices that, unlike Microsoft Windows for example, do not enable the storage media to be isolated and examined without the host device. The increased use of cryptography in storage devices is another factor preventing imaging in the conventional sense. The file-system cannot be read without the operation of the host operating system and the host operating system does not enable the storage media to be imaged.

Our belief is that the ACPO 2.2.4 proposal, to image rather than not, will increasing fail to be possible and so the practice of cryptographically hashing the entire media will be seen as the exception rather than the norm in large media. It may remain in the case of smaller media like memory sticks.

8.6.1.3 Availability

FCluster replicates DECs on, typically, three storage nodes. This is an arbitrary number based upon common practice in devices such as RAID and the Hadoop file system. Thus, we have redundancy to ensure availability.

One objective in the design of FCluster was to allow access to DECs for processing while not allowing the investigator to have complete access to copy the target file at will. FCluster can offer this. If, by means of access control, investigators are unable to directly mount an FCluster file-system in their own user space they can only gain access to the data by adding tasks to the system workflow table. If additions to the list were restricted to entries that called programs that were on an approved list then by controlling additions to an approved list, programs that could copy a whole file could be prevented from being run.

8.6.1.4 Non-repudiation

FCluster achieves effective non-repudiation by provision of a comprehensive logging system. This records all DEC ingestions, movements, unpacking and verification as well as any file-system *open* actions on the data.

8.6.2 Evaluate against ACPO v5, Dec 2007

In 3.5.2 we identified the ACPO Guidelines as key to the acceptance of assurance in digital forensics in the UK. Here we assess FCluster and FClusterfs against the relevant parts.

8.6.2.1 Principle 3

> *"An audit trail or other record of all processes applied to digital evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result"*

This is satisfied by the audit-trail within FClusterfs, provided it gathers enough information about the software used, including version number, FCluster should satisfy principle 3 of the APCO v5 guidelines.

8.6.2.2 ACPO 2.2.4

> *"In order to comply with the principles of digital evidence, wherever practicable, proportionate and relevant an image should be made of the device. This will ensure that the original data is preserved, enabling an independent third party to re-examine it and achieve the same result, as required by principle 3."*

This is discussed in section 8.6.1.2, as it is an issue of integrity.

8.6.2.3 ACPO 2.2.5

> *"This may be a physical / logical block image of the entire device, or a logical file image containing partial or selective data (which may be captured as a result of a triage process). Investigators should use their professional judgement to endeavour to capture all relevant evidence if this approach is adopted."*

This is the conflict between full imaging and selective data capture.

8.6.2.4 ACPO 2.2.7

> *"It is essential to display objectivity in a court of law, as well as the continuity and integrity of evidence. It is also necessary to demonstrate*

> *how evidence has been recovered, showing each process through which the evidence was obtained. Evidence should be preserved to such an extent that a third party is able to repeat the same process and arrive at the same result as that presented to a court."*

There are two aspects to this. Firstly, FCluster creates and maintains an audit trail of actions upon data stored within the system. The second is, largely, an issue with the tools used to recover information from the data and the decision of the investigator that it represents evidence. FCluster is designed to enable existing software to run and obtain access to data stored in FCluster without the need to any alteration to the software. This does mean that when software is run using FCluster it can be validated against the results of the same process running on non-FCluster platforms.

8.6.2.5 How robust is the database behind FClusterfs

Table 26 - inode table field updates by processing stage, on page 208, shows the progression of field updates as data is ingested into FClusterfs.

Updates to the MySQL database behind FClusterfs are at the heart of the acceptable movement of data through the assurance zones. At each stage, multiple checks are made against the table fields that should be completed if the data item had successfully achieved the previous assurance zone.

FClusterfs inherits a number of safeguards from MySQL. Firstly, MySQL is transactional in that when a number of Update or Insert commands are considered to be a batch to complete a job they are committed at the same time. In this way, partial transactions will always fail. Secondly, MySQL has a built in replication function that enables automatic duplication and failover functions. This also helps scalability and load balancing.

8.6.3 Against ISO 27037:2012

The evaluation of FCluster will focus on the creation of a Gap Analysis against ISO 27037:2013. From the sections identified in section 3.5.3, a Gap Analysis table can be drawn up.

ISO 27037:2012 defines the characteristics of a management system and the tools to identify, collect, acquire, preserve and analyse digital evidence. Not all sections of the standard are applicable when set against FCluster, which is limited to how the data is handled within a distributed processing environment. FClusterfs can be viewed as a management system to preserve digital evidence. The assessment is below in Table 32.

Clause	Title	Control Objective* *summarised from the ISO standard Text*	Is this control Required in FCluster* (SoA)	Assessment
5.3.2	Auditability	To make it possible for the actions of DEFR and DES to be evaluated.	Yes	All significant actions within FCluster are audited without being either too intrusive, and so affecting system response, or too detailed as to be unreadable.
5.3.3	Repeatability	Given the same set of circumstances, the outcome of an event should be identical or that any variation must be explainable.	Yes	This is not really a function of FCluster as programs run under the control of the host operating system. If the host operating system provides repeatability assurance then it is inherited by FCluster.
5.3.4	Reproducibility	Given the same digital evidence, the same results should be capable of being reproduced using DIFFERENT tools.	Yes	This is not really a function of FCluster as programs run under the control of the host operating system. If the host operating system provides reproducibility assurance then it is inherited by FCluster.

Clause	Title	Control Objective* *summarised from the ISO standard Text*	Is this control Required in FCluster* (SoA)	Assessment
5.3.5	Justifiability	That the DEFR or DES should be able to justify their actions.	No	Not Applicable
5.4.2	Identification	Each digital evidence container should be uniquely identifiable in an immutable way.	No	In 5.4.2, the standard uses ‘identify’ in the sense that the DEFR’s task is to locate digital data at a crime scene in order for it to be secured.
5.4.3	Collection	That the DEFR should gather together ready for data acquisition.	No	This is viewed as a ‘human’ function outside of FCluster.
5.4.4	Acquisition	That a suitable method of acquisition is chosen and then executed correctly. Acquisition should be verifiable.	Yes	This is not provided by FCluster, rather it comes from Jigsaw imaging which has already been assessed in section 8.3
6.8	Prioritising Collection and Acquisition	ISO 27037 highlights the need to prioritise the collection of more volatile data.	No	Not Applicable
6.9	Preservation of Potential Digital Evidence		Yes	This is really part of data prioritisation in Jigsaw imaging.

Clause	Title	Control Objective* *summarised from the ISO standard Text*	Is this control Required in FCluster* (SoA)	Assessment
5.4.5 & 7.1.4	Preservation	Potential evidence should be preserved to ensure its usefulness. It is important to protect the integrity of the evidence.	Yes	FCluster uses MD5/SHA1 cryptographic checksums to record the initial state of the data at acquisition and throughout its existence in the FCluster system. The integrity id repeatedly checked at scheduled intervals and just before use.
6.1	Chain of Custody	The DEFR and DES should be able to account for all the acquired data at the time it is within the custody of the DEFR and DES. The chain of custody should contain at least: • Unique evidence Identifier • Who accessed the evidence and the date time it took place • Who check the evidence in and out • Why the evidence was check out • Any unavoidable changes to the evidence	Yes	FCluster provides a comprehensive chain of evidence audit trail from the MySQL database. Specifically the audit table contains the relevant information.

Table 32 - Assessment against ISO 27037:2012

8.6.4 Against ISO 17025:2005

This is the ISO standard intended for application in a 'Testing facility' and so is not expected to be truly relevant. We include the assessment, in Table 33, to be comprehensive.

Clause	Title	Control Objective* *summarised from the ISO standard Text*	Is this control Required in FCluster* (SoA)	Assessment
5.2	Human Factors	Including competence, provision of on-going training, supervision, clearly defined tasks and responsibility.	No	Not Applicable
5.3	Accommodation and environmental conditions	The working environment should be adequate to support the analysis work undertaken. There should be environmental monitoring. There should be separation between areas that have incompatible activities. There should be good housekeeping in the laboratory.	No	Not Applicable

Clause	Title	Control Objective* *summarised from the ISO standard Text*	Is this control Required in FCluster* (SoA)	Assessment
5.4	Test and calibration methods and method validation	There should be appropriate testing and calibration of equipment that meets the needs of the customers. These should be based on international standards and/or the manufacture's own standards and procedures. Where formal methods do not exist, new methods should be established and validated. (cont) Methods used in analysis should be selected to meet the needs of the customers and be of a suitably high quality. There is provision for non-standard tests, in that they must be validated first. Selection of Methods (5.4.2 of ISO 17025) Laboratory-developed methods (5.4.3 of ISO 17025) Non-standard methods (5.4.4 of ISO 17025) Validation of Methods (5.4.5 of ISO 17025) Estimation of Uncertainty (5.4.6 of ISO 17025) Control of Data (5.4.7 of ISO 17025)	Perhaps?	Calibration does not seem to have a place in digital forensics. However, selection of methods does have it place but at the application program level, not at the operating system level.

Clause	Title	Control Objective* *summarised from the ISO standard Text*	Is this control Required in FCluster* (SoA)	Assessment
5.5	Equipment	The laboratory should have equipment that is at least adequate for the job. It should meet the required accuracy. Equipment should be uniquely identified. Equipment operated outside of the nominal levels should be subject to further conformance testing.	Yes	Not so much to do with FCluster as the equipment it is run on.
5.6	Measurement traceability	Equipment should be calibrated as per the manufacturer's and International System of Units' instructions. The laboratory should use reference standards.	No	Not Applicable
5.8	The handling of test and calibration items	Test and calibration sample should be handled and protected in a suitable manner.	No	Not Applicable
5.9	Assuring the quality of test and calibration results		No	Not Applicable
5.10	Reporting the Results		No	Not Applicable

Table 33 - Assessment against ISO 17025:2005

8.6.5 Against OAIS, ISO 14721:2012

The characteristics of the “Space data and information transfer systems — Open archival information system (OAIS) — Reference model” were described in section 4.3.7. This standard was identified by the reviewers of the paper submitted to DFRWS EU 2014. The design of FCluster was conducted unaware of this standard but does surprisingly well when assessed against it as shown in Table 34.

A system is said to conform if it “Supports the model of information described in 2.2” and “fulfils the responsibilities listed in 3.1” of ISO 14721:2012

Feature	Applicable	Present in FCluster	Notes
Producer	Yes	Yes	The DEFR
Archive	Yes	FClusterfs	
Consumer	Yes	Yes	The DES
Management	Yes	Yes	
SIP	Yes	Yes	Digital Evidence Containers
AIP	Yes	Yes	FCluster Storage Format
DIP	Yes	Yes	As a user available file

Table 34 - Applicability of ISO 14721:2012 against FCluster

FCluster certainly embodies the concepts described in ISO 14721:2014, section 2.2.1 – “Information Definition”, which is summarised in section 4.3.7.

The definition of the information data and meta-data as it is stored in FCluster would probably not conform to the Information Package Definition in ISO 14721:2014, section 2.2.2, but it is very close. In FCluster, at the stage that data is stored in Archive Information Package format (AIP), meta-data has been read from the digital evidence container (DEC or SIP) and is stored in the MySQL database used to organise data storage within the cluster. During the unpacking process, the meta-data is read from the DEC and written to a file with the same prefix but with a ‘.meta’ text attached as a suffix. There is no reason that a pointer to the metadata file could not be inserted into the database instead.

With hindsight, it would have been better to store data within FCluster in such a way that the original meta-data was held together with its data, which is actually the structure of the DEC created by Jigsaw imaging. This would improve resilience to corruption as a ‘stray’ DEC could be fully identified back to its source. OAIS allows for variation in the storage format of data as it is submitted, archived and disseminated. The downside is that it would require the data to be unpacked every time it is requested.

8.6.6 Against Daniel Ayers' criteria

From section 4.2.7, on page 77, in his proposal for a second-generation forensic analysis system, Ayers blurs the boundary between operating system and application program describing it all as 'the tool'. Some apply to one, some to both.

He identifies a series of metrics, shown in Table 35, by which a system should be judged

Characteristic	Meaning	Operating System	Application Program
Absolute Speed	measured by the time elapsed from start to finish	X	X
Relative Speed	a ratio between the read speed of the storage media and the processing speed	X	X
Accuracy	the proportion of results returned that are correct		X
Completeness	the proportion of evidence found		X
Reliability	that the tool does not crash and recovers from errors	X	X

Table 35 - Ayers' general criteria

And then a series of requirements, shown in Table 36, for second generation tools

Characteristic	Meaning	Operating System	Application Program
Parallel Processing	The tool must be able to use the computational resources of many separate processors (i.e. processors that do not share main memory or I/O bus bandwidth) so as to be capable of improved absolute and relative speed.	X	
Data Storage and I/O Bandwidth	The tool must support a fault tolerant, high performance and scalable data storage medium	X	
Accuracy and Reliability	The tool must be designed and coded to provide a high level of assurance that analysis results will be correct and software operation free from error under all circumstances.		X
Auditability	Source code for forensic analysis functions should be available for independent review by a qualified third party.	X	X
Repeatability	The tool must support the automation of all analysis functions and processes, except those where interactive human involvement is unavoidable.		X

Characteristic	Meaning	Operating System	Application Program
Data Abstraction	The tool must provide high-level abstrac-tions for at least the following types […] Common data formats,		X

Table 36 - Ayers' criteria for second-generation tools

Selecting the characteristics that apply to the Operating System or Middleware, we see that FCluster certainly achieves parallel processing, scalable storage and auditability. Reliability is achieved by replication of data. The two criteria of absolute and relative speed fall into the same problem areas as previously identified by Roussev in that they are comparative with existing benchmark systems like FTK. All we can say is that prioritising high value evidence data improves delivery to the processing system, see Figure 106 on page 230, and that once in a distributed system, blocking of smaller jobs no longer hampers the processing of larger jobs.

We feel that we have satisfied Ayers' requirement that apply to the Operating system and Middleware.

8.7 Evaluation against the project objectives

We now return to section 1.3.5 and evaluate the original individual objectives of the research.

8.7.1 Objective 1 was:

> *"To derive a set of requirements to enable the development of a distributed management system specifically suited to forensic investigation."*

In chapters 3, 4 we undertook extensive research into the standards that influence the practice of digital forensics, primarily in the UK but not ignoring the heavy influence of the practice in the USA. We considered current implementations practices and identified the presence of a chain of evidence in systems currently used for forensic analysis that failed to hold its integrity when employed in a distributed environment. This was distilled as a set of design requirements in chapter 5 and in Appendix A1.

8.7.2 Objective 2 was:

> *"To evaluate some prominent existing distributed management systems and assess their suitability to implement a prototype distributed forensic system."*

In sections 4.3 and 4.4, we considered BOINC, Hadoop, HTCondor and a number of distributed file system such as AndrewFS, GlusterFS and PVFS. We found the each design stems from a design criteria with objectives to solve specific problems. General purpose distributed systems could be employed in forensic processing but most lack either immediacy, such as HTCondor which sends data away to be processed on whatever platform is available at the time, or by the distributed file system which often splits files into strips or chunks to increase delivery speed or reliability. Our conclusion was that because none had been designed with the specific need of forensic processing in mind, none was effective in this use case.

8.7.3 Objective 3 was:

> *"To classify existing forensic investigation tools and assess the likelihood of running them in a distributed environment and so to derive a standard for new tools intended to run within a distributed environment"*

In section 4.3.5, rather than looking at specific software we instead looked at the way in which they operate in terms of batch mode that may, or may not, require user intervention. Because FCluster presents as a modified file system, using FUSE, a large amount of existing software will run unaltered. In fact, much software can be run across the cluster to leverage the distributed processing without any modification. The only exception is the type often found in Microsoft Windows environments where the code is integrated with the user interface. This software will run but cannot exploit the distributed processing. This is the same as existing software that runs on a single host and reads data from a file server.

8.7.4 Objective 4 was:

> *"To develop a robust design of a middleware framework to support processing digital forensic tools in a distributed environment."*

In chapters 3, 4 and 5 we believe that we have conducted an extensive design assessment and have developed a middleware framework that supports a number of fundamental principles in digital forensics. In chapter 8, we assessed this system against several of the principle standards used in digital forensics and, on balance, believe it to have succeeded.

8.7.5 Objective 5 was:

> *"To evaluate the prototype system using representative case data."*

Aware of Garfinkel's observation we wanted to use realistic data in our assessment. We used two disks one of 256GB and one of 500GB. These were several times larger than the largest available in Garfinkel's Corpus. Arguably, we could have used even larger disks. The University of South Wales uses 2TB disk as a standard media in all of its machines for students but these do not store any user data, instead user data is stored on a data storage warehouse facility. The disk we used were the largest, most realistic available.

8.8 Evaluation against the research hypothesis

From 1.3.4 the hypothesis is:

> *"It is possible to facilitate the timely handling of large-scale digital evidence for professional computer forensic investigations, whilst still maintaining an appropriate chain of evidence, through the design of a suitable acquisition and processing methodology, implemented within distributed middleware architectures."*

We started this research knowing, from examples in Hadoop, HTCondor and GRID, that distributed processing usually only yields its potential after the data has been distributed. In so many systems, the penalties incurred by distributing the data negate the gains of being able to process data in parallel. We saw it in forensic systems like FTK when setup of distributed processing and this also is true of FClusterfs if taken in isolation.

However, by extending the principle of concurrency out to the acquisition process, we were able to deliver DECs as soon as they became available from Jigsaw imaging. This happens during the acquisition stage at the same time as the image is being created. As a result, FCluster is able to start distributing and therefore processing, well ahead of the conventional practice of linear imaging and processing on a single host. This was achieved without significant impact on the overall time spent imaging using a more conventional linear approach.

Further to this, we proposed a priority system that pushed the acquisition of potentially higher value evidence to the fore. The scoring system used can be customised by the DEFR to suit the case profile.

We evaluated the component parts of the system. We can demonstrate by providing a prototype that Jigsaw imaging works and the non-linear nature has little effect on overall processing times. Further, we demonstrated, by explanation, that prioritisation does achieve its objectives. We evaluated the chain of evidence within FClusterfs from the point of view of an auditor when faced with an audit against an ISO standard where the auditor would extend the SoA and Gap Analysis conducted by the organisation being audited and consider it to have passed an audit.

In our objectives, we stated that we would need to derive a standard for new tools intended to work in this environment. The implementation of an assurance framework by using a FUSE file system negated this entirely. We believe no additional skills or programming practices would be needed to access the distributed processing power via FCluster.

We believe that FCluster does provide an appropriate chain of evidence for distributed forensic processing of data and that Jigsaw imaging combined with prioritisation does provide timeliness to the overall process.

As Ayer's observed, as to whether this is accepted by the ultimate customer, the legal profession remains to be answered.

8.9 Published papers

During the course of this research there were three papers published. The main paper, "Information Assurance in a distributed forensic cluster", was presented at Digital Forensics Research Workshop EU 2014, in Amsterdam and subsequently published in Digital Investigation.

- 2008 – Pringle N., Sutherland I., "Is a Computational Grid a Suitable Platform for High Performance Digital Forensics?", 8th European Conference on Information Warfare and Security, University of Plymouth, UK.
- 2014, Pringle, N., Burgess, M., "Information Assurance in a distributed forensic cluster". Digital investigation (11), S27-S35, doi:10.1016/j.diin.2014.03.005
- 2014, Digital Forensics Research Workshop US 2015, Denver, USA. Short presentation

8.10 Conclusions

In chapter 8, we reported on our evaluation of the project. We described the evaluation strategy and proceeded to evaluated it on Jigsaw imaging, Prioritisation, the Digital Evidence Container and FClusterfs. Sometimes we were able to conduct quantitive, timed, evaluation and sometimes use a qualitative approach by assessing its conformance to various standards we had identified in chapters 1, 3 and 4.

The chapter was concluded with the associated publications and an assessment of this work within the thread of research over the past decade.

9 Conclusions, Applications and Further Work

9.1 Introduction

In the previous chapters, 3 and 4, we presented both the non-technical and the technical background to the problems associated with the increased volume of data that need be analysed for forensic investigators. This led on to a design brief, in chapter 5, in which we stated the solution requirements. Chapter 6 described FCluster and its component parts. In chapters 7 and 8, we tested and evaluated our design.

We set out to provide a solution to what is probably the biggest problem facing digital forensics since it started two decades ago. It has been said that all of the gains of the last two decades could be lost if our investigators simply cannot keep up with the volume of data they are expected to process in a manner acceptable to the legal system. The consequence of continued use of existing architectures will result in our inability to process huge volumes of data and so either create pressure to lower our standards of evidence in digital matters or simply the adoption of a triage approach where 'lower level' crimes are not investigated and prosecuted.

Here we draw together all of the work from the previous chapters and present our conclusions and our deliverables.

9.2 Summary of our analysis

The increase in evidential data was first apparent about 15 years ago. In response to this, Roussev proposed a system called DELV, recounted in section 4.3.2, that involved moving data from a data store to the processing nodes. This is the architecture used in HTCondor and is currently in use in Accessdata's FTK Forensic version. Our own first attempt at solving this problem used Grid computing to mimic Roussev's work and proved that it was not too bloated to be discounted from the solution set but was certainly not rich in features that would make it acceptable for use within the legal process. These designs highlighted the conflict between a conventional forensic image of media and the overhead or moving data around for processing. In the intervening years, there have been a few proposals for systems but few have progressed past the stage of a broad outline of a proposal. The National Forensic Institute in Holland and Lightbox in the United States have both used Hadoop in their designs. Neither has published their work and so it is only possible to speculate on their success. Because of the problems associated with the lack of random access within the Hadoop File system, we speculate that they are extracting data using very high-powered single PCs and storing forensic information from the source data in some form in the Hadoop architecture. We surmise that they must base further processing on this abstracted data. We feel there is a danger that in abstracting it is possible to loose information present in the original. This approach also makes re-processing of historic data rather inefficient as the originals would most likely need to be retrieved from some long-term archival system.

9.3 Our Offering

FCluster is the first example of a middleware specifically designed for the distributed storage and processing of large amounts of data that addresses the need for processing in a forensically sound manner. The data acquisition, ingestion, storage and processing techniques were designed with forensic soundness as a primary objective from the start.

FCluster is designed as a series of interlocking zones that, although connected, have clear divisions and embody a progression of data as it passes through a series of checks and controls that provide assurance that the data acquired at imaging time is the same as is available to whatever program is tasked to process it.

9.4 Contributions to Research

Prior to this project, no one had addressed the problem of massive processing of data for forensics in a way that tried to acknowledge the need for assurance of the chain of evidence in digital forensics. Roussev had created DELV and Ayers had created a broad criterion by which a system would be judged. We introduced chain of evidence to their work.

9.5 As part of a research thread

Despite being identified by several surveys as one of the most pressing problems within digital forensics, there has been relatively little research on this particular problem area. Quick & Choo's review (2014) of the whole subject area confirms this with only eight papers directly identified as mentioning “Distributed Parallel”. The eight included 2 concerned with GPUs and and 2 review papers. This leaves just 4 papers working directly on distributed parallel processing.

Roussev's original proposal for DELV a decade ago was mirrored in a Grid in our own paper in 2008. A year later, it was developed upon by Ayers' publication in 2009. Ayers' article included plans for system he intended to build but it seems he never did. There have been three attempts at similar solutions in the last 5 years. Accessdata have provided distributed processing in their FTK product since version 4. However, this still relies upon a central file server that stored whole images of data. Two other projects have been identified, the first by Lightbox and the second by the National Forensic Institute in the Netherlands. Both seem to be based on a design where meta-data is extracted or created from the original material and the meta-data is stored within an Hadoop based system, perhaps with HBase. There is little information available about either of these systems. The most recent publication is was ours in 2014.

FCluster takes the original observations by Roussev and Richard and then overlays Ayers' more detailed thoughts about the characteristics of such a system and how it should be judged. It brings in the influences of the more recent thinking about Digital Evidence Containers that did not exist when Roussev created DELV and tied them all together with a file system that included an inherent chain-of-evidence. In terms of data storage, we took the notion from Hadoop to have local processing of data at the point of storage, and the ability to modify a file system with FUSE to create access to this distributed storage for legacy software.

FCluster successfully draws together several recent threads into a solution that, we believe, provides an original approach to a serious problem.

9.6 Immediate Applications – Impact

9.6.1 Patent Application

This work is current to subject of a patent application shown in appendix D and a proposal for Horizon 2020 funding over the next few years.

9.7 Future Work

There are a number of avenues of research that could develop from this work.

9.7.1 On the question of priorities

In this work, we introduced a Bayesian scorecard approach to choosing which data would be best to process as having a high potential value of yielding actual evidence. This leads to the question, is it possible to identify certain targets by the characteristics of the crime being investigated. It seems little work has been done within digital forensics on the behaviour of suspects when they commit certain types for crime. Work by Hong et al., Horsman and Rogers are seen in isolation. The behavioural characteristics of suspects are well researched in other areas. This work repeats Rogers' call for more data to be generated on criminal behaviour and their personalities in a cyber-environment.

9.7.2 On the question of job queuing

Connected with the previous issue is the question that with a limited amount for processing power is better to process 10,000 smaller files or a handful of larger files.

9.7.3 A portable cluster based on small 'System On a Chip' boards.

What has become clear during the research his that this architecture can be applied both at the macro and the micro scale. It was imagined that FCluster would be manifest as a cluster of i7 PCs but during the course of the research, a number of "System On a Chip" (SoC) devices have become available.

The Benchmarks shown in Table 37 give an indication of the relative processing power and the relative electrical power consumption.

Processor	Floating Point MIPS (Whetstone) **per CPU**	Integer MIPS (Dhrystone) **per CPU**	Typical Power Consumption
CubieTruck Cortex A7	454 (x 1)	1856 (x 1)	0.5 W (x 1)
Atom	1700 (x 3)	2600 (x 1)	2.0 W (x 4)
Intel i3	2160 (x 4)	6751(x 3)	80W (x 160)
Intel i7	2980 (x 6)	10112 (x 5)	150-250 W (x 400)
Xeon X3220	14567 (x 32)	57752 (x 31)	250W (x 500)
	Buffered Disk Read MB/s	Cached Reads MB/s	Typical Power Consumption
WD 2" Hard Disk on SATA	75	412	6w
Crucial-M SSD on SATA	115	424	1w
WD 2" hard Disk on USB	27	397	6w
Crucial-M SSD on USB			1w

Table 37 - Relative Benchmarks

During 2014, Cubietruck boards had an A7, 2-core processor and cost about £80. Only 6 months later the Raspberry Pi 2 has an Armv7 A20 4-core processor and costs less than £30.

As an addition to the LiveDVD we could also offer a small system based upon a Cubietruck device.

This poses the possibility of a scene of crime portable distributed system that could regain the "Golden Hour" identified in section 4.7.

References

AccessData Corporation, 2009. Divide & Conquer: Overcoming Computer Forensic Backlog through Distributed Processing and Division of Labor. Available at: http://www.techrepublic.com/resource-library/whitepapers/divide-conquer-overcoming-computer-forensic-backlog-through-distributed-processing-and-division-of-labor/ Last Accessed: 03-04-2015.

AccessData Corporation, 2014. Available at: http://www.accessdata.com Last Accessed: 03-04-2015.

ACPO, 2014. ACPO Guide to Digital Evidence v5. Available at: http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf Last Accessed: 03-04-2015.

Apache, 2014. Apache HBase, 5.12. Joins. Available at: http://hbase.apache.org/book/joins.html Last Accessed: 03-04-2015.

ARC Group, 2014. ProDiscover® Forensic Edition. Available at: http://www.arcgroupny.com/products/prodiscover-forensic-edition/ Last Accessed: 03-04-2015.

Ayers, D., 2009. A second generation computer forensic analysis system. *Digital Investigation*, Volume 6, pp.S34–S42 doi:10.1016/j.diin.2009.06.013.

BBC, 2014a. Child abuse image investigation leads to 660 arrests. *BBC Web Site*. Available at: http://www.bbc.co.uk/news/uk-28326128 Last Accessed: 03-04-2015.

BBC, 2009. Dozens arrested in burglary raids. *BBC Web Site*. Available at: http://news.bbc.co.uk/1/hi/england/west_yorkshire/8084192.stm Last Accessed: 03-04-2015.

BBC, 2006. Eleven charged over ’bomb plot’. *BBC Web Site*. Available at: http://news.bbc.co.uk/1/hi/uk/5271998.stm Last Accessed: 03-04-2015.

BBC, 2014b. Police ’overwhelmed’ by number of child abuse images. *BBC Web Site*. Available at: http://www.bbc.co.uk/news/uk-29470001 Last Accessed: 03-04-2015.

BBC, 2014c. TV technology for the 2020 Olympics. *BBC Web Site*. Available at: http://www.bbc.co.uk/news/technology-29789468 Last Accessed: 03-04-2015.

Black Lab, 2015. System Imager. Available at: http://www.os4online.com/2013/04/forking-remastersys-and-state-of-os4.html Last Accessed: 03-04-2015.

Brancatelli, A., 2014. MySQLfs. Available at: http://andrea.brancatelli.it/category/tech/mysqlfs-tech/ Last Accessed: 30/04/2015.

Breitinger, D. F. Baier H. White, 2014. On the database lookup problem of approximate matching. *Digital Investigation*.

Budzier, A. & Flyvbjerg, B., 2011. One in six IT projects ends up ‘out of control.

Carnegie Mellon University, 2015. *What is Andrew?*, {Carnegie Mellon University}. Available at: http://www.cmu.edu/corporate/news/2007/features/andrew/what_is_andrew.shtml Last Accessed: 03-04-2015.

Carrier, B., 2005. *File System Forensic Analysis*, Addison Wesley.

Carrier, B., 2015. Sleuthkit. Available at: http://www.sleuthkit.org Last Accessed: 01/03/2015.

Casey, E.,Katz, G. & Lewthwaite, J., 2013. Honing digital forensic processes. *Digital Investigation*, Volume 10(2), pp.138–147 doi:10.1016/j.diin.2013.07.002.

CDESF, D., 2006. Survey of Disk Image Storage Formats, Version 1.0. Available at: http://www.dfrws.org/CDESF/survey-dfrws-cdesf-diskimg-01.pdf Last Accessed: 03-04-2015.

Cellebrite, 2014. Cellebrite. Available at: http://www.cellebrite.com/ Last Accessed: 03-04-2015.

Cohen, M.,Garfinkel, S. & Schatz, B., 2009. Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow. *Digital Investigation*, Volume 6, pp.S57–S68 doi:10.1016/j.diin.2009.06.010.

Cohen, M.,Garfinkel, S. & Shatz, B., 2014. Affuse. Available at: http://www.forensicswiki.org/wiki/Affuse Last Accessed: 03-04-2015.

ComputerWorld, 2005. DOD seized 60TB in search for Iraq battle plan leak. Available at: http://www.computerworld.com.au/article/15477/dod_seized_60tb_search_iraq_battle_plan_leak/ Last Accessed: 03-04-2015.

Cornell University, 2014. Rule 702. Testimony by Expert Witnesses. Available at: http://www.law.cornell.edu/rules/fre/rule_702 Last Accessed: 03-04-2015.

Coulouris, G. et al., 2012. *Distributed Systems, Concept and Design. 5th Edition* M. Horton, ed., Addison-Wesley.

dd, 2014. dd. Available at: http://standards.ieee.org/findstds/standard/1003.1-2008.html Last Accessed: 03-04-2015.

Dean, J. & Ghemawat, S., 2004. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI '04: 6th Symposium on Operating Systems Design and Implementation*. USENIX Association.

Dictionary.com, 2014. Available at: http://Dictionary.com http://www.swgde.org.

Electronic Frontier Foundation, 2014. Privacy: Searching and Seizing Computers. Available at: https://ilt.eff.org/index.php/Privacy:_Searching_and_Seizing_Computers Last Accessed: 03-04-2015.

Enslow, P.H., 1978. What is a "Distributed" Data Processing System? *Computer*, 11(1), pp.13–21.

Al Fahdi, M.,Clarke, N. & Furnell, S., 2013. Challenges to digital forensics: A survey of researchers & practitioners attitudes and opinions. In *Information Security for South Africa, 2013*. IEEE, pp. 1–8.

Farmer, W. D. Venema, 2005. *Forensic Discovery* 2nd ed., Addison-Wesley Professional Computing Series.

Farrell, P., 2014. Digital memory imaging system and method. , (2503600).

FBI, 2014a. A Brief History of the FBI. Available at: http://www.fbi.gov/about-us/history/brief-history/brief-history Last Accessed: 03-04-2015.

FBI, 2014b. RCFL Program Annual Report. Available at: http://www.rcfl.gov/annual-reports/rcfl-program-annual-report last Accessed: 03-04-2015.

Ferris, T.L.J., 2009. On the Methods of Research for Systems Engineering. In *7 th Annual Conference on Systems Engineering Research 2009*.

Flament, R., 2013. LoggedFS - Filesystem monitoring with Fuse. Available at: http://loggedfs.sourceforge.net/ Last Accessed: 03-04-2015.

Forensic Science Regulator, 2011. Codes of Practice and Conduct. , 1.0. Available at: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/351197/The_FSR_Codes_of_Practice_and_Conduct_-_v2_August_2014.pdf Last Accessed: 03-04-2015.

Forensic Science Regulator, 2014. *Codes of Practice and Conduct Appendix: Digital Forensic Services, FSR-C-107 Issue 1*, H M Government. Available at: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/351220/2014.08.28_FSR-C-107_Digital_forensics.pdf Last Accessed: 03-04-2015.

forensicswiki, 2014a. ASR Data's Expert Witness Compression Format. Available at: http://www.forensicswiki.org/wiki/ASR_Data%27s_Expert_Witness_Compression_Format Last Accessed: 03-04-2105.

forensicswiki, 2014b. Encase image file format. Available at: http://www.forensicswiki.org/wiki/Encase_image_file_format Last Accessed: 03-04-2015.

forensicswiki, 2014c. Forensics File Formats. Available at: http://www.forensicswiki.org/wiki/Category:Forensics_File_Formats Last Accessed: 03-04-2015.

Foster, I., 2005. Globus Toolkit Version 4: Software for Service-Oriented Systems. In Springer-Verlag LNCS 3779.

Freeman, D.L. R. Edward; Reed, 1983. Stockholders and Stakeholders: A new perspective on Corporate Governance. *California Management Review*, 25(3), pp.88–106.

FUSE, 2014. FUSE. Filesystem in Userspace. Available at: http://fuse.sourceforge.net/ Last Accessed: 03-04-2105.

Garfinkel, S. et al., 2006. Advanced Forensic Format: an Open Extensible Format for Disk Imaging. In M. Olivier & S. Shenoi, eds. *Advances in Digital Forensics II*. IFIP Advances in Information and Communication. Springer New York, pp. 13–27. Available at: http://dx.doi.org/10.1007/0-387-36891-4_2 Last Accessed: 03-04-2015.

Garfinkel, S., 2010. Digital forensics research: The next 10 years. *Digital Investigation*, Volume 7, pp.S64–S73 doi:10.1016/j.diin.2010.05.009.

Garfinkel, S., 2014. DigitalCorpora.org. Available at: http://digitalcorpora.org/ Last Accessed: 03-04-2015.

Garfinkel, S., 2012. Lessons learned writing digital forensics tools and managing a 30TB digital evidence corpus. *Digital Investigation*, Volume 9(Suppliment), pp.S80–S89 doi:10.1016/j.diin.2012.05.002.

Garfinkel, S., 2009. Providing Cryptographic Security and Evidentiary Chain-of-Custody with the Advanced Forensic Format, Library, and Tools. In *International Journal of Digital Crime and Forensics (IJDCF)*.

Gluster.org, 2015. Gluster. Available at: http://www.gluster.org/ Last Accessed: 03-04-2015.

Gogolin, G., 2010. The Digital Crime Tsunami. *Digital Investigation*, Volume 7(1), pp.3–8 doi:10.1016/j.diin.2010.07.001.

Griffith, S.B., 1963. *The Art of War*, Oxford University Press, New York.

Guidance Software, 2014. EnCase. Available at: https://www.guidancesoftware.com/ Last Accessed: 03-04-2015.

H M Goverment, 2014. Human Rights Act 1998. Available at: http://www.legislation.gov.uk/ukpga/1998/42/contents Last Accessed: 03-04-2015.

H M Government, 2014. Forensic Science Regulator. Available at: https://www.gov.uk/government/organisations/forensic-science-regulator Last Accessed: 03-04-2015.

Harbour, N., 2014. dcfldd (2002-2006). Available at: http://dcfldd.sourceforge.net/ Last Accessed: 03-04-2015.

Hicks, T.,Kirkland, D. & Halcrow, M., 2013. eCryptfs, a cryptographic stacked filesystem for Linux. Available at: http://dubeyko.com/development/FileSystems/eCryptfs/ecryptfs.pdf Last Accessed: 03-04-2015.

HistoryofInformation.com, 2014. Probably the First U. S. Legislation against Computer Crimes (1978). Available at: http://www.historyofinformation.com/expanded.php?id=3888 Last Accessed: 03-04-2015.

Hockney, D., 2015. Available at: http://www.hockneypictures.com/current.php Last Accessed: 03-04-2015.

Home Office, 2014. List of UK police forces. Available at: http://www.police.uk/forces/ Last Accessed: 03-04-2015.

Hong, I. et al., 2013. A new triage model conforming to the needs of selective search and seizure of electronic evidence. *Digital Investigation*, Volume 10, p.doi:10.1016/j.diin.2013.01.003.

Horsman, G.,Laing, C. & Vickers, P., 2011. A Case Based Reasoning System for Automated Forensic Examinations. In *PGNET 2011 The 12th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, 27-28 June, 2011*.

Horsman, G.,Laing, C. & Vickers, P., 2014. A Case-Based Reasoning Method for Locating Evidence During Digital Forensic Device Triage. *Decision Support Systems*.

Instant-Grid, 2013. Instant Grid. Available at: http://www.instant-grid.org/ Last Accessed: 03-04-2015.

Intel, 2014. Intel® Solid-State Drive Data Center Family for PCIe. Available at: http://www.intel.co.uk/content/www/uk/en/solid-state-drives/intel-ssd-dc-family-for-pcie.html Last Accessed: 03-04-2015.

International League of Polygraph Examiners, 2014. Polygraph/Lie Detector FAQs. Available at: http://www.theilpe.com/faq_eng.html Last Accessed: 03-04-2015.

Interxion, 2015. The Laws of Technology: Driving Demand in the Data Center. Available at: http://www.interxion.com/blog/the-laws-of-technology-driving-demand-in-the-data-center/ Last Accessed: 03-04-2015.

ISO, 2012a. *ISO 14721:2012 Space data and information transfer systems — Open archival information system (OAIS) — Reference model*, ISO. Available at: http://www.iso.org/iso/catalogue_detail.htm?csnumber=57284 Last Accessed: 03-04-2015.

ISO, 2013. *ISO 27001:2013 "Information technology— Security techniques — Information security management systems — Requirements*, ISO. Available at: http://www.iso.org/iso/home/standards/management-standards/iso27001.htm Last Accessed: 03-04-2015.

ISO, 2012b. *ISO 27037:2012 Information technology — Security techniques — Guidelines for identification, collection, acquisition, and preservation of digital evidence*, ISO. Available at: http://www.iso.org/iso/catalogue_detail?csnumber=44381 Last Accessed: 03-04-2015.

ISO, 2014. *ISO 7498-2:1989 Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*, ISO. Available at: http://www.iso.org/iso/catalogue_detail.htm?csnumber=14256 Last Accessed: 03-04-2015.

ISO, 2012c. *ISO 9001:2008 Quality management*, ISO. Available at: http://www.iso.org/iso/catalogue_detail?csnumber=46486 Last Accessed: 03-04-2015.

ISO, 2004. *ISO/IEC 13335-1:2004 Information technology – Security techniques – Management of information and communications technology security – Part 1: Concepts and models for information and communications technology security management*, ISO. Available at: http://www.iso.org/iso/catalogue_detail.htm?csnumber=39066 Last Accessed: 03-04-2015.

ISO, 2012d. *ISO/IEC 17020:2012 Conformity assessment – Requirements for the operation of various types of bodies performing inspection*, ISO. Available at: http://www.iso.org/iso/catalogue_detail?csnumber=52994 Last Accessed: 03-04-2015.

ISO, 2005. *ISO/IEC 17025:2005 General requirements for the competence of testing and calibration laboratories*, ISO. Available at: http://www.iso.org/iso/catalogue_detail.htm?csnumber=39883 Last Accessed: 03-04-2015.

Keller, M., 2014. ROfs. Available at: https://github.com/cognusion/fuse-rofs Last Accessed: 03-04-2015.

Van der Knaap, N., 2013. Backlog in digital forensics: Is justice being done? Available at: http://leidenlawblog.nl/articles/backlog-in-digital-forensics-is-justice-being-done Last Accessed: 03-04-2015.

Kruse, W. & Heiser, J., 2001. *Computer Forensics: Incident Response Essentials.*, Addison Wesley.

Kryder, M. & Kim, C., 2009. After Hard Drives—What Comes Next? In IEEE Transactions on Magnetics. IEEE, pp. 3406–3413. Available at: http://isites.harvard.edu/fs/docs/icb.topic1195323.files/After_Hard_Drives.pdf Last Accessed: 03-04-2015.

Laney, D., 2001. 3D Data management: Controlling Data Volume, Velocity, and Variety. Available at: http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf Last Accessed: 03/04/2015.

Legal Information institute, 2014. Fourth Amendment. Available at: http://www.law.cornell.edu/wex/fourth_amendment Last Accessed: 03-04-2015.

Linux Kernel Newbies, 2007. Linux 2.6.14. Available at: http://kernelnewbies.org/Linux_2_6_14#line-36 Last Accessed:03-04-2015.

Livny, M., 1983. *The Study of Load Balancing Algorithm for Decentralized Distributed Processing Systems*. Sicientific Council of the Weizmann Institute of Science.

Lustre, 2015. Lustre. Available at: http://wiki.lustre.org/index.php/Main_Page Last Accessed: 03-04-2015.

Lyon, B., 2013. Your Refrigerator Just Attacked Your Bank. Seriously, It Could Happen. *Wired Magazine*. Available at: http://insights.wired.com/profiles/blogs/your-refrigerator-just-attacked-your-bank-seriously-it-could Last Accessed: 03-04-2015.

Magritte, R., 1929. La Trahison des images.

Marziale, L., 2009. *Advanced Techniques for Improving the Efficacy of Digital Forensics Investigations*. University of New Orleans.

Marziale, L.,Richard III, G. & Roussev, V., 2007. Massive threading: Using GPUs to increase the performance of digital forensics tools. *Digital Investigation*, Volume 4S, pp.S73–S81 doi:10.1016/j.diin.2007.06.014.

Meijer, R.J., 2014. Generic forensic zip project. Available at: http://www.nongnu.org/gfzip/ Last Accessed: 03-04-2015.

Message passing Interface (MPI), 2014. *MPI: A Message-Passing Interface Standard 1.0*, University of Tennessee, Knoxville, Tennessee. Available at: http://www.mcs.anl.gov/research/projects/mpi/indexold.html Last Accessed: 03-04-2015.

Metropolitan Police, 2014. Big Wing Operation targeting burglary and theft nets over 630 arrests. Available at: http://content.met.police.uk/News/Big-Wing-Operation-targeting-burglary-and-theft-nets-over-630-arrests/1400023761809/1257246745756 Last Accessed: 03-04-2015.

Microsoft, 2014a. Common Internet File System. Available at: http://technet.microsoft.com/en-us/library/cc939973.aspx Last Accessed: 03-04-2015.

Microsoft, 2014b. *Scheduling Disk Defragmenter to run regularly*, Microsoft. Available at: http://windows.microsoft.com/en-gb/windows/schedule-regular-disk-defragmenter#1TC=windows-7 Last Accessed: 03-04-2015.

MIT, 2014. libcurl - the multiprotocol file transfer library. Available at: http://curl.haxx.se/libcurl/ Last Accessed: 03-04-2015.

Mockler, R.J., 1970. *Readings in Management Control.*, New York: Appleton-Century-Crofts.

Moore, G.E., 1965. Cramming More Components onto Integrated Circuits. *Electronics*, pp.114–117.

MySQL, 2014. MySQL and the ACID Model. Available at: http://dev.mysql.com/doc/refman/5.6/en/mysql-acid.html.

National Institute of Science & Technology, 2005. *Digital Data Acquisition Tool Test Assertions and Test Plan (Draft 1 of Version 1.0, Nov 10, 2005)*, National Institute of

Science and Technology. Available at: http://www.nist.gov/itl/ssd/cs/cftt/upload/DA-ATP-pc-01.pdf Last Accessed: 03-04-2015.

National Police Improvement Agency, 2007. *The Golden Hour. A competent investigator knows that time is crucial to an investigation.*, {National Police Improvement Agency}. Available at: http://www.college.police.uk/en/docs/Assessment_Guide_05-31_2G3.pdf Last Accessed: 07-11-2014.

NIST, 2014. Disk Imaging. Available at: http://www.nist.gov/itl/ssd/cs/cftt/cftt-disk-imaging.cfm Last Accessed: 03-04-2015.

Office of National Statistics, 2012. Family Spending. Available at: http://www.ons.gov.uk/ons/rel/family-spending/family-spending/index.html Last Accessed: 03-04-2015.

Palmer, G., 2001. DTR-T001-01 Technical Report. A Road Map for Digital Forensic Research. In *Digital Forensics Workshop (DFRWS), Utica, New York*.

Parsonage, H., 2009. Computer Forensics Case Assessment and Triage. Available at: http://computerforensics.parsonage.co.uk/triage/ComputerForensicsCaseAssessmentAndTriageDiscussionPaper.pdf Last Accessed: 03-04-2015.

Pollitt, M., 1995. Computer Forensics: An Approach to Evidence in Cyberspace. In *Proceeding of the National Information Systems Security Conference*. pp. 487–491.

Pollitt, M., 2013. Triage: a practical solution or admission of failure. *Digital Investigation*, Volume 10, pp.P87–88 doi:10.1016/j.diin.2013.01.002.

Pringle, N., 2004. A prototype of a client/server forensic program using XML Unpublished MSc work, University of Glamorgan.

Pringle, N. & Sutherland, I., 2008. Is a Grid a Suitable Platform for High Performance Digital Forensics? In *The 7th European Conference on Information Warfare and Security*.

PVFS, 2015. Parallel Virtual File System. Available at: http://www.pvfs.org/ Last Accessed: 03-04-2015.

Quick, D. & Choo, K.-K.R., 2014. Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digital Investigation*, p.doi:10.1016/j.din.2014.09.002.

Reith, G. M. & Carr C. & Gunsch, 2002. An Examination of Digital Forensic Models. *International Journal of Digital Evidence*.

Richard, G. & Roussev, V., 2006. Digital forensics tools: the next generation. *Digital Crime and Forensic Science in Cyberspace. Idea Group Publishing*, pp.75–90.

Richard, G.,Roussev, V. & Marziale, L., 2007. Forensic discovery auditing of digital evidence containers. *Digital Investigation*, Volume 4(2), pp.88–97 doi:10.1016/j.diin.2007.04.002.

Richard III, G. & Roussev, V., 2005. Scalpel: A Frugal, High Performance File Carver. *Digital Investgation*.

Robson, B.A., 2013. CurlFtpFS - An FTP filesystem based on Curl and FUSE. Available at: http://curlftpfs.sourceforge.net/ Last Accessed: 03-04-2015.

Rogers, M.,Seigfried, K. & Tidke, K., 2006. Self-reported computer criminal behavior: A psychological analysis. *Digital Investigation*, Volume 11(S3), pp.S116–S120 doi:10.1016/j.diin.2006.06.002.

Roussev, V.,Quates, C. & Martell, R., 2013. Real-time digital forensics and triage. *Digital Investigation*, Volume 10, pp.158–167 doi:10.1016/j.diin.2013.02.001.

Roussev, V. & Richard III, G., 2004. Breaking the performance wall: The case for distributed digital forensics. In *Proceedings of the 2004 Digital Forensics Research Workshop*.

Sandberg, R. et al., 1985. Design and Implementation or the Sun Network Filesystem.

Schatz, B., 2007. *Digital Evidence Representation and Assurance.* PhD thesis, Queensland University of Technology. Available at: http://eprints.qut.edu.au/16507/ Last Accessed: 03-04-2015.

Schmuck, F. & Haskin, R., 2002. GPFS: A Shared-Disk File System for Large Computing Clusters. *Proceedings of the FAST 2002 Conference on File and Storage Technologies*. Available at: https://www.usenix.org/legacy/events/fast02/full_papers/schmuck/schmuck.pdf Last Accessed: 03-04-2015.

Scientific Working Group on Digital Evidence, 2005. ASCLD Glossary Definitions: v1.0. Available at: http://www.swgde.org Last Accessed: 03-04-2015.

SGI, 2012. SGI UV. Available at: http://www.sgi.com/products/servers/uv/index.html Last Accessed: 03-04-2015.

Shaw, A. & Browne, A., 2013. A practical and robust approach to coping with large volumes of data submitted for digital forensic examination. *Digital investigation*, Volume 10, pp.P116–128 doi:10.1016/j.diin.2013.04.003.

Soubra, D., 2012. The 3Vs that define Big Data. Available at: http://www.datasciencecentral.com/forum/topics/the-3vs-that-define-big-data Last Accessed: 03-04-2015.

Stafanov, K.S. & Brijeski, T., 2013. Remastersys 3.0.4-1. Available at: http://www.remastersys.com Last Accessed: 01-10-2013.

Systemation, M., 2015. .Xry. Available at: https://www.msab.com/ Last Accessed: 03-04-2015.

Texas Instruments, 2014. Jack Kilby. Available at: http://www.ti.com/corp/docs/kilbyctr/jackstclair.shtml Last Accessed: 03-04-2015.

Thain, D.,Tannenbaum, T. & Livny, M., 2005. Distributed computing in practice: The Condor experience. *Concurrency and Computation: Practice and Experience*, 17(2-4), pp.323–356.

The Library of Congress, 2014. METS Metadata Encoding and Transmission Standard. Available at: http://www.loc.gov/standards/mets/ Last Accessed: 03-04-2015.

The National Center of Victims of Crime, 2014. Evidence Retention Laws. Available at: http://victimsofcrime.org/docs/default-source/dna-resource-center-documents/evidence-retention-check-chart-9-5.pdf?sfvrsn=2 Last Accessed: 03-04-2015.

The Register, 2014. Kryder’s law craps out: Race to UBER-CHEAP STORAGE is OVER. Available at: http://www.theregister.co.uk/2014/11/10/kryders_law_of_ever_cheaper_storage_disproven/ Last Accessed: 03-04-2015.

The Register, 2012. UV 2: RETURN of the 'Big Brain'. This time, it's affordable. Available at: http://www.theregister.co.uk/Print/2012/06/19/sgi_uv_2000_xeon_super/ Last Accessed: 03-04-2015.

The University of California, 2014. Berkeley Open Infrastructure for Network Computing (BOINC). Available at: http://boinc.berkeley.edu/ Last Accessed: 03-04-2015.

Turner, P., 2006. Selective and intelligent imaging using digital evidence bags. *Digital Iinvestigation*, Volume 3, pp.59–64 doi:10.1016/j.diin.2006.06.003.

Turner, P., 2005. Unification of digital evidence from disparate sources (digital evidence bags). *Digital Investigation*, Volume 2(3), pp.223–228 doi:10.1016/j.diin.2005.07.001.

United Nations Office on Drugs & Crime, 2013. *Comprehensive Study on CyberCrime February 2013*, United Nations.

US Department of Justice FBI, 2012. Regional Computer Forensics Laboratory Annual Report For Fiscal Year 2012. Available at: http://www.rcfl.gov/annual-reports Last Accessed: 03-04-2015.

Walter, C., 2005. Kryder's Law. Available at: http://www.scientificamerican.com/article/kryders-law/ Last Accessed: 03-04-2015.

Weil, S.A., 2007. *CEPH: RELIABLE, SCALABLE, AND HIGH-PERFORMANCE DISTRIBUTED STORAGE*. UNIVERSITY OF CALIFORNIA, SANTA CRUZ. Available at: http://ceph.com/papers/weil-thesis.pdf Last Accessed: 03-04-2015.

White, T., 2012. *Hadoop: The Definitive Guide, 3rd Edition, ISBN: 9781449328917*, O'Reilly Media, Inc.

Wikipedia, 2013. Hard disk drive. Available at: http://en.wikipedia.org/wiki/Hard_disk_drive Last Accessed: 03-04-2015.

Wikipedia, 2014a. History of computer clusters. Available at: http://en.wikipedia.org/wiki/History_of_computer_clusters Last Accessed: 03-04-2015.

Wikipedia, 2014b. History of supercomputing. Available at: http://en.wikipedia.org/wiki/History_of_supercomputing Last Accessed: 03-04-2105.

Woods, K.,Lee & Garfinkel, S., 2011. Extending Digital Repository Architectures to Support Disk Image Preservation and Access. In *JCDL'11*.

Woods, Kam et al., 2011. CREATING REALISTIC CORPORA FOR SECURITY AND FORENSIC EDUCATION. In *ADFSL Conference on Digital Forensics, Security and Law*.

Appendixes

A1 – Problems, Solutions, Features and Evaluation.

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
1	Linear imaging is ‘blocking’ in nature and needs A to complete before B can initiate	Speed	As we found, conventional Linear Imaging does not easily align with parallel processing, as it needs imaging to finish before any processing can take place. We seek an alternative method of acquiring the potential evidence set in a timely manner so that distributed storage and processing can be fully utilised.	Current Forensic Imaging Formats Growth of Media File Sizes Parallelisation The end of the Image	Replace linear imaging with a new paradigm – Jigsaw Imaging. Jigsaw Imaging reads data off the source in Digital Evidence Containers that are processed as they are created rather than waiting until the whole image is complete. However, while Jigsaw Imaging is reading each cluster needed to build the DECs, it is also writing out the sectors that form the cluster to a corresponding sector on a target drive.	**6.3**	**8.3**

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
2	Matching data transfer speed against processing speed to form a well-balanced system	Speed	Data transfer over gigabit Ethernet is typically 100MB/s. AccessData's FTK can process about 2MB/s on a single core of an i7 processor. If the data is stored on a central file-server, it is possible to supply data to occupy only 50 cores or about seven i7 hosts with each host having 8 cores each.	3.5.5 Hadoop Roussev 2013	The FClusterfs FUSE File system exploits 'data locality' by implementing an architecture where whole files are stored on a remote host and a method of initiating a task on that remote host is provided	**6.7.2.5**	Future Work 9.7
3	We do not want to waste time processing data that is unlikely to yield evidence.	Speed	Not all files are equally interesting in terms of their likelihood to host evidence. If we spend time processing low value data, we are wasting time. It would be better if the system directed the processing power to the most effective application of the power.	ISO 27037 - 6.8 Prioritising Collection and Acquisition	Prioritisation is based on a user defined points score system that attempts to attach a potential evidential value to each file. Files that score higher values are processed first, with lower value files processed later or may even be ignored.	**6.5**	**8.3.7**

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
4	Complex Processing tasks sometimes block simple tasks	Speed	Large jobs, for example creating thumbnails on ever key-frame in a video can be very time consuming and may block the progress of processing other data.		As data is replicated across the cluster, a blocked process can be run on a host that holds one of the data replicas. Big tasks still run exclusively to their conclusion but out on a suitably allocated processing host.	Not implemented in the prototype	Future Work 9.7

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
5	Digital Forensics needs meta-data in addition to the typical data needed by other users.	Extensibility	When we copy data, understandably, it does not usually carry the meta-data about its original location on the source media. When we take a forensic image, this meta-data is inherent in the structure of the image. If we adopt DECs, we need to collect and store the meta-data so that that it is available to the host investigative system.	**3.5.2** ACPO v5 - 2.2.5 Partial or Selective data, preservation of relevant evidence. ACPO v5 - 2.2.7 Integrity of evidence	The FClusterfs FUSE File system holds the extended meta-data required in a forensic investigation. A digital forensic investigation is surely the only activity that requires that the original file data and location of the original file data on the storage media meta-data be recorded. However, other types of users do require similar types of information relevant to their domains. This meta-data is stored in the core database and so cannot be separated from the rest of the data stored on a cluster.	**6.7.2.12**	8.4.5

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
6	Access to raw data and evidence needs to be controlled.	Confidentiality	If the system is to be accessed by multiple agencies then a corresponding granular access control system needs to be in place to support this. Investigators, including software, will need to be able to identify themselves so that the system can use this information to allow and restrict access accordingly, even down to individual file level. The Take-Grant security system used in NTFS is a good example of this fine-grained access control. From the forensic computer system design, this poses particular problems when the exchange of evidence and the information derived from investigations have to cross jurisdictions. Ideally, there should be a separation between the data itself and the results derived from processing it. In a similar way to the release of a cryptographic hash database of illegal data, typically photographs, allows sharing of a key characteristic of the data but not the data itself.	**3.7** Cross media forensics **4.4** Multi Agency Access	The FClusterfs FUSE File system controls all access to the data. We solve this by employing a database table containing access identification, authentication and asset management. Granular access to data will allow the system to satisfy the requirements to maintain of privacy for stakeholders	**6.7.2.9**	**8.6.1** **8.6.1.1**

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
7	We should not waste the programming efforts of the last 20 years.	Accessibility	During the course of the last 20 years a large corpus of software has been developed and it would be desirable if as much of this as possible is still able to run on the solution system without alteration.	**4.3.6** **4.3.1.8** **4.3.5**	The FClusterfs FUSE File system controls how the data is presented to users. The contents of Digital Evidence Containers present as 'ordinary' files that require no further unpacking by the application program.	**6.7**	**8.6.1.3**
8	It can be a difficult to learn the programming paradigm used by a new architecture.	Accessibility	Writing new software should not require too many new skills to exploit the advantages of distributed processing. The need to understand and adopt a new paradigm for software development and data access is a barrier to future developments.	**4.3.1**	The FClusterfs FUSE File system unpacks the Digital Evidence Containers on demand. They will present as conventional files to any application program that needs access. There is no need for additional procedures or libraries to be added to new application programs. Programmers can use any programming language.	**6.7.2**	**8.6.1.3**

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
9	Handling large numbers of storage volumes from different media can be very overpowering.	Scalability	The RCFL Reports (US Department of Justice FBI 2012) over the last few years has shown media size and the number of individual items of media are increasing within a single investigation. Ideally, a system should be sufficiently scalable that it could be sufficiently scalable to allow several hundreds of file-systems to be processed.	3.2	The FClusterfs FUSE File system stores file and file-system metadata in an SQL database allows the middleware to address large numbers of files. Filters can be applied to increase or decrease the evidence presented to a processing host.	**6.7.2.3**	**8.6.6**

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
10	We do not want data to be intercepted and copied during transmission	Confidentiality	The nature of the data held within this system requires high levels of confidentiality to be maintained at all times. Data transmission is a particular risk point. Although there are wider issues of Governance, there is no technical reason that processing data at forensic standards should be restricted to a closed environment, as is the current practice.	**3.5.3** ISO 27037 - 6.9.4	The FCluster allows wide area, multi organisational access to data. All communication links are encrypted. Encryption should ideally be suitably strong, typically AES-256 or 3DES, and 'End-to-End' with the data being encrypted during transmission and storage on the media. SSH is an obvious choice. The design can use protocols such as HTTPS and SSH in all communications links. However, PKI, digital certificates and tunnelling of application level protocols over SSH may introduce a level of complexity in a prototype. It is better to avoid this. Subsequently we may need to employ an inferior protocol in the proof of concept implementation. These are open standards.	**6.7.2.6**	**8.6.1.1** **8.6.3**

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
11	We must maintain confidentiality when disposing of media used on the system.	Confidentiality	If our forensic processing used cloud services it could offer us an elastic facility able to accommodate variable work-loads but there is a risk that residual data traces may remain even after the most rigorous cleansing. Any data stored on remote media must be encrypted to maintain its confidentiality after use and closedown of the cloud facility.	**3.5.3** Extension of ISO 27037 - 6.9.4 Cloud Storage and Processing	The FClusterfs FUSE File system is built upon the eCryptfs fuse file system that seamlessly encrypts and decrypts data on the media. Encryption is strong, typically using AES-256 standard.	**6.7.2.6**	**8.6.3**
12	The system must be transparent to external audit	Accessibility	We can expect that this system will be subject to external challenge within the legal system and validation by organisations such as NIST to establish its suitability in the legal process. Complexity makes it difficult to argue and establish the system's assurance.	**4.3.7** ISO 14721:2012 OAIS	The system uses Open Standards and Open Source Code practices allowing the system to be more accessible for auditing.	**6.2**	**8.6.1.3**

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
13	Monitoring of the movement and processing of data must take place without the application programmer's involvement.	Auditability, Traceability, Accountability	The system should record all processing actions in enough detail to satisfy an audit but this should be achieved without application programs being aware of this monitoring and not requiring special coding practices.	**3.5.3** ISO 27037 - 5.3.2 **3.5.2** ACPOv5, Principle 3	The FClusterfs FUSE File system is built upon the Loggedfs fuse file system that records all file interactions without modification of the application programs. The level of monitoring strikes a balance between the inevitable speed reduction that will occur if auditing is too intrusive and the detail of information collected. All 'File Open' action is recorded but not individual cluster reads within files. The audit trail identifies data such as date, time, user, application program used and host ID. Data is tracked from its authority to be captured through to its destruction or removal from the system.	**6.7.2.8**	**8.6.1.4** **8.6.3**

	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
14	Users/Investigators must not be able to alter data stored on the system	Integrity	Data access should be read-only but further than this, it may be desirable to prevent uncontrolled replication of the data.	**3.5.2** ACPO v5 Imaging 2.2.4	The FClusterfs FUSE File system is built upon the ROfs fuse file system that eliminates the code that allows application programs to write to files. In effect creating a Read-Only file system.	**6.7.2.7**	**8.6.1.2** **8.6.2.2**
15	The integrity of data needs to be guaranteed	Integrity	Loss of integrity of the data leaves it worthless for presentation in court as evidence.	**3.5.3** ISO 17025	Using cryptographic hashes such as MD5, SHA-1 and SHA-2 establishes an initial record of the state of the data as it leaves the original media. Including the cryptographic hash within the Digital Evidence Container's design and its later ingestion into the SQL data as part of the file meta-data means it can be accessed periodically and used to verify the integrity of the file.	**6.7.2.13**	**8.6.1.2**

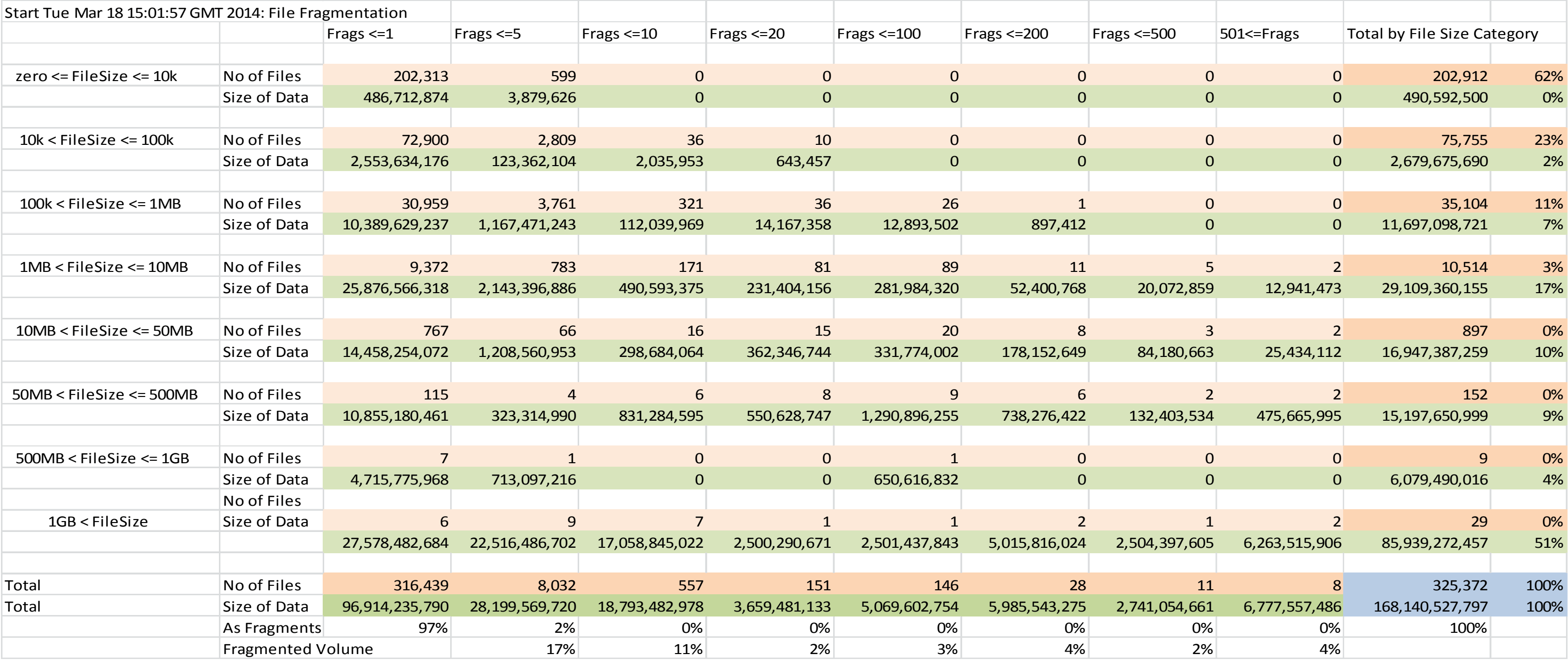
	Problem	Principle	Why is it a problem?	Origin From Chapters 1, 3 and 4	Solution	Feature Description Chapter 6	Evaluation Section Chapter 8 and 9
16	Running one instance of a program on one instance of a PC raises the chance that undetected mistakes and errors may occur	Verifiability	Usually because of the complexity of a system, running what appears to be the same hardware/software combination does not always perform in exactly the same way. This puts processing assurance in question.	**3.5.3** (ISO 17025) **3.5.3.3** (ISO 27037 5.3.3 and 5.3.4)	Processing assurance increases if a task runs on a variety of differing machines. It is of advantage to write the system in a manner that can run on as wide a range of host operating systems and hardware as possible.	Not implemented in the prototype	Not implemented in the Prototype
17	Evidence needs to be unique identification.	Integrity	Confusion over the identity of a DEC and subsequently its source would be a severe impairment to its claim to integrity assurance. When there are more than 100,000 files in what was a single forensic image we need a rigorous method of uniquely identifying each DEC.	**3.5.3.3** (ISO 27037 - 5.4.2)	Each DEC has a unique identification that cannot be lost or changed without detection. It should also be able to allow the original media to rebuild it as a replica of the original.	**6.5**	**8.6.1.2**
18	Involvement of the investigator to trigger routine tasks slows the overall task.	Speed	Involving the investigator/operator in triggering tasks forms a block on the progress of the overall task.		Involving the investigator/operator in triggering tasks forms a block on the progress of the overall task.	Not implemented in the prototype	Not implemented in the Prototype

A2 - General Requirements of Distributed Systems

Principle	Description
Access Transparency:	where local and remote resources are access using identical operations.
Location transparency;	enables resources to be accessed without knowledge of their location.
Concurrency transparency;	enables several processes to operate concurrently using shared resources without interference between them.
Failure transparency;	enables the concealment of faults. This is often solved by replication or redundancy.
Heterogeneity;	service interfaces should be designed in a way so that clients and server software can be implemented for different operating systems and hardware.
Replication transparency;	enables multiple instances of resources to be used to increase reliability and performance without the knowledge of the users or their applications.
Mobility transparency;	allows the movement of resources within a system without affecting the operation of users and programs.
Performance transparency;	allows the system to be reconfigured to improve performance as loads vary.
Scaling Transparency;	allows the system to be reconfigured in scale without affecting the operation of the users.

A3 - Data Processing Priority

File Count, Data Size and Fragmentation



Start Tue Mar 18 15:01:57 GMT 2014: File Fragmentation											
		Frags <=1	Frags <=5	Frags <=10	Frags <=20	Frags <=100	Frags <=200	Frags <=500	501<=Frags	Total by File Size Category	
zero <= FileSize <= 10k	No of Files	202,313	599	0	0	0	0	0	0	202,912	62%
	Size of Data	486,712,874	3,879,626	0	0	0	0	0	0	490,592,500	0%
10k < FileSize <= 100k	No of Files	72,900	2,809	36	10	0	0	0	0	75,755	23%
	Size of Data	2,553,634,176	123,362,104	2,035,953	643,457	0	0	0	0	2,679,675,690	2%
100k < FileSize <= 1MB	No of Files	30,959	3,761	321	36	26	1	0	0	35,104	11%
	Size of Data	10,389,629,237	1,167,471,243	112,039,969	14,167,358	12,893,502	897,412	0	0	11,697,098,721	7%
1MB < FileSize <= 10MB	No of Files	9,372	783	171	81	89	11	5	2	10,514	3%
	Size of Data	25,876,566,318	2,143,396,886	490,593,375	231,404,156	281,984,320	52,400,768	20,072,859	12,941,473	29,109,360,155	17%
10MB < FileSize <= 50MB	No of Files	767	66	16	15	20	8	3	2	897	0%
	Size of Data	14,458,254,072	1,208,560,953	298,684,064	362,346,744	331,774,002	178,152,649	84,180,663	25,434,112	16,947,387,259	10%
50MB < FileSize <= 500MB	No of Files	115	4	6	8	9	6	2	2	152	0%
	Size of Data	10,855,180,461	323,314,990	831,284,595	550,628,747	1,290,896,255	738,276,422	132,403,534	475,665,995	15,197,650,999	9%
500MB < FileSize <= 1GB	No of Files	7	1	0	0	1	0	0	0	9	0%
	Size of Data	4,715,775,968	713,097,216	0	0	650,616,832	0	0	0	6,079,490,016	4%
	No of Files										
1GB < FileSize	Size of Data	6	9	7	1	1	2	1	2	29	0%
		27,578,482,684	22,516,486,702	17,058,845,022	2,500,290,671	2,501,437,843	5,015,816,024	2,504,397,605	6,263,515,906	85,939,272,457	51%
Total	No of Files	316,439	8,032	557	151	146	28	11	8	325,372	100%
Total	Size of Data	96,914,235,790	28,199,569,720	18,793,482,978	3,659,481,133	5,069,602,754	5,985,543,275	2,741,054,661	6,777,557,486	168,140,527,797	100%
	As Fragments	97%	2%	0%	0%	0%	0%	0%	0%	100%	
	Fragmented Volume		17%	11%	2%	3%	4%	2%	4%		

Table 38 - File Sizes and Fragmentation

From the previous Table 38 - Information Summary

325,372 files total, 68,048 known 'ok' files against NIST database of 32,122,997 records

96GB, 316,000 files, 97% not fragmented,
8GB, 8,000 files, 2% fragmented less than 5 fragments,
7GB, 8 files, 1% fragmented more than 5 fragments

3GB, 275,000 files, 85% of files by number with less than 100k (25 clusters)

A4 - The Design Argument

- Hard disks are getting bigger, there is more data to capture and analyse
- This takes longer and is hampering investigation and so prosecution of crime
- There is an urgent need to reduce the timescale of collection, processing and analysis

Q: There is a data overload problem in digital forensics. What are the alternative solutions?

Several things can be done.

Q: Why don't we wait for Intel to develop faster **computers?**

A: Because the criminals are buying them as well and thereby generate more data, which takes longer to process and the status quo continues. What we need is leverage over the criminals that they don't want to get or can't get. This also applies to 'Intel is bringing out an amazing chip next year. It's a revolution'.

Q: We could use a mega PC with 64 cores **and SSD RAID 5 arrays**

A: These are disproportionately expensive.

Q: Why not use GPUs?

A: GPUs need specific programming. This would mean either throwing away all those existing tools or at least not being able to use them fast. Also this still requires the data to be read from the disk at 100MB/s for hard disk, 500MB/s for SSD

Q: Can we do data processing triage?

A: You mean selectively ignore stuff that might be evidence of guilt or innocence. Is this really a good approach?

Q: What about "Data Reduction"?

A: Research in "Data Reduction" is aimed at discarding 'low value' data and focusing on data that is likely to yield evidence. Of all the approaches, so far this sounds most promising. There is no doubt that a DLL file is less likely to yield evidence than a JPG. In fact, investigators have always done this.

Q: Forensics isn't about investigating loads of data just very clever analysis.

A: It may be true for clever consultants with PhDs. You tell that to the run of the mill DC in a local digital forensic lab when they have yet another 2TB drive to analyse and only have 1 day's pay in their budget to spend on that analysis.

Q: Could new clever algorithms be developed?

A: Clever algorithms often end up with more processing not less. A law of diminishing returns applies. In addition, there would need to be hundreds of victories, one for each technique, not just one.

Q: What about Distributed Processing, like the sort of thing that has been so effective in Big Data?

A: This could give provide huge leverage in the processing available to law enforcement over their suspect's capabilities. Distributed parallel processing is scalable and provides cheap bang for the buck. M^{2}COTS are understandable, can run existing legacy software, and does not exclude GPUs or clever algorithms. Hey, let's do it!

Q: Does AccessData FTK have distributed processing already?

A: FTK uses a central file server to store all the images. When the system is used, a bottleneck develops in the file store network connection. If a 4TB image is stored on a file server and sections of it are copied out to half a dozen workstations, it results in copying 4TB along a 1 gigabit Ethernet cable. It will take about 11 hours. That is not too bad but that is just one investigation at a time. If there are several going on at the same time it becomes very slow.

This was the approach taken by Vassil Roussev in 2004 in Breaking the Performance Wall and DELV.

What is need is distributed storage and processing just like Hadoop. This is based on distributing the data storage and processing the data in locality. Distributed processing has been around for many years. Distributed storage and processing is what makes Hadoop and Map/Reduce so effective.

The trouble is that distributed storage is a direct contradiction to 'imaging' which is a very centralised technique.

Q: So what about imaging. It is one of the founding principles of best practice in digital forensics.

A: I cannot help but conclude that 'forensic imaging' has a limited life expectancy, With more data being stored 'in the cloud' and more evidence within individual investigations coming from difference sources, it's hard to see how we can continue to successfully collect it in 'one big file'. In addition, in the US, there have been challenges on the issue of privacy when an image is taken and data is collected that is not 'owned' by the suspect and is not relevant to the case being investigated. Taking an image is increasingly being seen as a violation of the Human Right to privacy. I am not the first see this. Guidance software has introduced their 'Logical Evidence' file format in which individual files, and collections of files, can be stored. AFF format has always had this facility.

Q: So why not just take a selection of files, AFF bag them, copy them up on a memory stick, then copy them onto a PC and have a look?

A: Even though we may not have an image we should at least try to uphold some process that would be considered forensically sound from outside the domain.

Q: What is forensic soundness?

A: Well that a good question. Up to now, we have treated digital evidence in much the same way as analogue evidence. It is collected and 'bagged' as an 'image' with an identifiable tag. A device, such as a hard disk or memory stick, has been considered to be one single item of evidence. It may contain many artefacts that will yield much information but in terms of the chain of evidence it is a singular item. When it is moved, or copied, a record is kept. Typically, an evidence store is a locked room in a Police station. In most cases, it is a single place. A computer has been treated in the same way. It is considered a single place or perhaps the evidence is considered to be in the custody of an investigator. In most cases this has been manifest in the use of either EnCase or FTK on a workstation running Microsoft Windows and, more recently, connected to a local area network with a central file storage system. I think this breaks down when 'the' computer is a distributed cluster of, perhaps, thousands of computers. If we do not move to a distributed approach we will just loose more ground to the increase in data to be analysed. To move to a distributed system we need to improve chain of evidence within the computer system.

Q: Hadoop is flavour of the month! Then why don't you use Hadoop?

A: Hadoop works best with a small number of very large files rather than a very large number of small files. This makes it sound as if it is ideal for processing forensic image files. However, Hadoop uses a storage technique in common with many distributed file systems – striping. To attain much higher data read speeds, distributed file systems often spread the contents of a single file across many storage hosts. This is rather like RAID but on a grand scale. Distributed file systems stripe in two ways. Ceph, for example, stripes like RAID 5, in that it spreads the bits within a byte, across different storage hosts. On the other hand, Hadoop File system keeps complete bytes but chops up whole files into chunks, so sections of a file are held together on a storage host but the rest of the file is spread across many other storage hosts. This is ideal when processing sections of sequential files that reside within the same chunks but does not work with random access files that require the file pointer to jump around the file. It the pointer stays within the chunk it works but this cannot be guaranteed. If a large file with a random access is processed on an Hadoop file system there would be loads of data transfer from other storage nodes as each part of the whole file was needed. This would neutralise any advantage gained from the MapReduce algorithm that is based upon locality in processing. Unfortunately, a forensic image is in effect a very large random access file. Hadoop holds its directory database in RAM on a Namenode and so has a finite limit, as it will fill RAM. If the individual files from within an image are stored on Hadoop then the maximum number of files will soon be reached as the directory is filled well before the storage space is filled. Hadoop does have a compression and packing facility, much like ZIP, but this involves unpacking data which extends the processing time. Catch 22.

Q: So what are Lightbox and the Dutch NFI doing?

A: They won't give details of their work so I can't be specific but I can guess from one or two comments that were said when I visited the NFI in 2014. I believe they are using multiple big PCs with 64 cores to process the original images and then store the results within Hadoop. They then process the results using Hadoop and HBase etc. It is difficult to estimate their budget but with a permanent, devote development staff of about 35 it must be approaching £25m pa. They can afford this because they are a government organisation. They say that their objective is to provide the whole of Dutch Law enforcement, and associated agencies, with a national system. The Netherlands is also the host for The International Court of Justice, The International criminal Court, Europol and Eurojust. These are all hosted in The Hague, hence the city has become known as the World's Legal Capital and The Netherlands has aspirations and a budget to match.

Q: So what's wrong with this?

A: It a question of layers of abstraction. The NFI approach seems to be that they process the original data to create a set of data which they store in Hadoop hBase. That data is then processed to obtain 'information'. However, if the original data extraction is wrong or can be improved by re-processing then they would have to return to the original files and reprocess. It would be much better if the original data was stored in a more usable form that would allow it to be processed at any time.

Q: Why not jump on the Hadoop bandwagon and start modifying it?

A: To adopt Hadoop is to start with a simple assumption that because Hadoop is distributed and processes data, it must be good at the distributed processing of all data. This does not acknowledge that there is variation in data types and that these variations lead to differing optimum solutions. In Breaking the Performance Wall, Vassil Roussev rejected existing systems as ‘bloated’ because they have to cater for as wide a usage as possible. However, starting from scratch to build an entire operating system and file system would be equally prohibitive. I think Google’s approach to build middleware upon a well-established base, ie Linux, is the most likely to yield results. Hadoop has been in development for a decade. HDFS and the NameNode software is about 6.5 million lines of code. Although this might be a good route in a commercial environment, it is not appropriate for my purpose.

Q: What are the different approaches?

A: The fundamental problem is keeping track of data objects within a wide area storage system and organising its processing across many hosts. Keeping track of the filenames sounds like a database problem. This is about storing regular delimited or fixed length records about lists of files. In fact, it is a directory system. Should this be sql or non-sql? Well the file contents will vary (3V) but the characteristics of the filenames and meta-data will not.

- We could store an entire forensic image, as a single field, within a database. One record per image with one BLOB field as a massive binary. – Clean but not really practical when an image could be many tetrabytes.
- We could store the entire filesystem, including file contents, in a database. Some fields to store metadata and one BLOB to store the file contents. – Not unreasonable. The database system would then make the choice of the media location of the data. Not sure this is a good idea.

We could store the directory entry, including metadata, and a then pointer to the file in the file system. The file could be an image, a DEC or a file. – Better, we still retain control of the actual data location but the directory, held in the SQL can be replicated to load balance.

It sounds like SQL. SQL relational database technology is very mature. Existing software like Oracle, MySQL, Postgres and MariaDB can handle billions of records and the new Innodb indexing makes it incredibly fast. These databases have a whole host of facilities like fall-over, replication and distributed storage then FCluster would just inherit. The only problem is that ever application program would need to be modified or written to be database aware. This means amending existing programs and new programs having to adopt new practices. Any amendments to the database schema in future versions would have to be applied very carefully to be retrospective. This is not a very good solution. What we need is a layer that’s completely transparent but it is everywhere across the system. A distributed file system designed specifically for forensic data processing.

Q: So what are the specifics of the approach you have taken?

A: I’m proposing three things. Firstly, I’ve looked at Hadoop and a number of existing distributed file-systems and designed a distributed file-system specifically to provide facilities which can be used to ensure forensic soundness in the handling of data. Secondly, I've proposed an improved imaging process by basing it around digital evidence bags while still retaining backwards compatibility by producing an image at the same time. Thirdly, I've introduced prioritisation of data acquisition and processing. This means that data with a high potential is delivered to the distributed system in a forensically sound manner before the imaging process has ended.

Q: More specifically….

A: Corresponding to the HDFS File-system within Hadoop, I have designed a file system called FClusterfs. Mimicking the way HDFS is implemented as a middleware between application program and the native EXTx file system, FClusterfs is implemented as a middleware. I have chosen to use the FUSE file-system technique rather than write from scratch as with Hadoop. The key difference is that individual files within a file system are stored as complete, whole, files across a multitude of remote locations. The proximity of two file names in a directory listing does not imply any proximity in their storage locations. I have devised a novel technique for imaging forensic data which tracks through the media which I have called Jigsaw Imaging. It does not work by cluster or LBA sequence, but by tracking through the files. As it does this it simultaneously writes each block to a corresponding location on an output device to form an image. Finally the order that the files are processed within Jigsaw Imaging is determined by a priority system which assigned a simple numeric value to each file, sorts them all, and works through the list in descending order of potential value.

Q: Hasn't all this been done before?

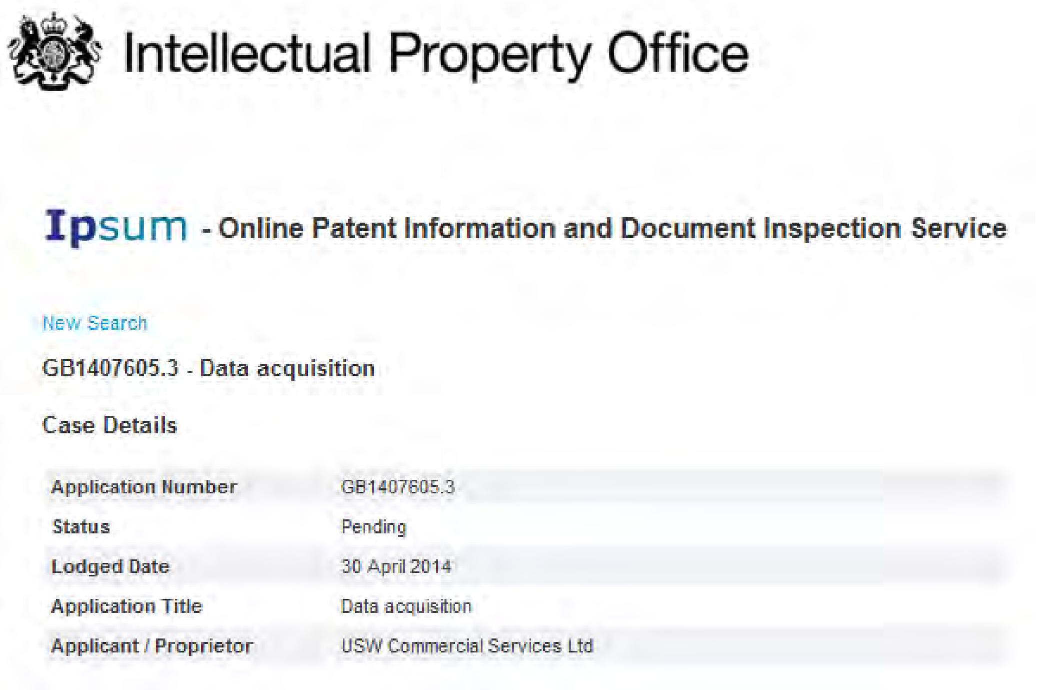
A: Although there are similar developments none are quite suitable. I guess the main evidence that they do not exist is that if they did, then why is the Digital Forensic Community struggling to cope with the problem of data volume?

Q: How can we see this?

A: I have implemented this in three forms. Firstly, as a series of virtual machines hosted within VMWare Desktop. Secondly, as a LiveCD which can be used to boot a host without affecting it's installed operating system. The LiveCD attempts to mount any local hard disk and use it for storage but any data is stored in an AES-256 encrypted form so nothing is left in an interpretable form when the system reboots to its normal function. Thirdly, it is implemented as a cluster of six Cubietruck SoC minicomputer boards. Each has a 500GB hard disk, 2 cores, 2GB of RAM and a 1 Gigabit Ethernet connection. This is controlled by a 10" touch screen. This mini-cluster runs Debian Wheezy operating system.

A5 - Patent Application – 1407605.3

https://www.ipo.gov.uk/p-ipsum/Case/ApplicationNumber/GB1407605.3



Intellectual Property Office

Ipsum - Online Patent Information and Document Inspection Service

New Search

GB1407605.3 - Data acquisition

Case Details

Application Number	GB1407605.3
Status	Pending
Lodged Date	30 April 2014
Application Title	Data acquisition
Applicant / Proprietor	USW Commercial Services Ltd

Claims

1. A method for copying data from an original resource to a plurality of target resources, the method comprising:

prioritising the data copying, based on the prioritising, at least some of the data to a plurality of target resources.

2. A method according to claim 1, the method further comprising:

reading, from the original resource, data indicative of the structure of data in the target resource;

copying, from the original resource to the plurality of target resources, data representing a directory of the original resource;

prioritising, based on the data representing the directory, files of the original resource for copying;

copying, based on the prioritising, at least some of the files of the original resource to the plurality of target resources.

3. A method according to claim 1 or 2, wherein at least one target resource, as a result of the copying, comprises a forensic image of the original resource.

4. An apparatus for copying data from an original resource to a plurality of target

resources, wherein the apparatus is configured to carry out the method of any of claims 1 to 3.

5. A method of analysing data originating from an original resource, the method comprising:

receiving data from the original resource;

distributing the received data to one or more second resources; and

processing the data at the one or more second resources.

6. A method according to claim 5, wherein receiving the data further comprises:

receiving data indicative of metadata associated with the received data.

7. A method according to claim 5 or 6, wherein one or more copies of the data is stored at one or more of the second resources.

8. A method according to claim 7, wherein the method further comprises:

verifying the data stored at one of the second resources based on data stored at one or more other resources.

9. A system for analysing data originating from an original resource, wherein the system is arranged to carry out the method of any of claims 5 to 8.

A6 - FOI Response Details from Gwent Police

All requests for information to Gwent Police under the Freedom of Information Act 2000 are routinely published on the Gwent Police web site (http://corporate.gwent.police.uk/foi/foiresponses/). There have been two that provide us with useful information relevant to this research.

Cybercrime

Added: 25 September 2014 **Category:** Crime
Reference: 17145

Disclosure

Q1. What training is available to:

(a) officers

(b) civilian staff to improve digital skills and for the purpose of investigating cybercrime and cyber-enabled crime?

A1. **Cyber training package for open source internet investigation is being delivered to selected front line Detectives and Intelligence officers to improve the Force response to cyber enabled crime investigation.**

This training is the pre cursor to more enhanced training to in house experts in relation to sophisticated cyber- crime related offences.

Q2. How many full time equivalent:

(a) officers

(b) civilian staff have completed training in cyber or digital skills in the last 12 months up to August 2014?

A2. Hi-Tech Crime Unit (HTCU) Staff have attended the following courses
a) Three Police Officer have attended Encase Transition to Version 7 course
b) One Support Staff has attended Encase Transition to Version 7 course
c) Three Support Staff attended Specialist Encase Mackintosh Examination course
d) Two Support Staff attended Access Data FTK Beginners course
e) Two Support Staff attended XRY Examination course (Beginners)
f) Two Support Staff attended XRY Examination course (Intermediate)

Q3. How many:

(a) officers

(b) civilian staff are assigned to the cybercrime (or equivalent) unit?

If there is no unit specifically focussed on cybercrime, how many full time equivalent:

(a) officers

(b) civilian staff are dedicated to investigating cyber and cyber-enabled crime?

A3. **There are no dedicated cyber/cyber-enabled crime officers or support staff within the HTCU, however all HTCU members can be assigned to deal with such matters**

Q4. How many investigations have been initiated by the cybercrime (or equivalent) unit in the last 12 months up to August 2014?

A4. **Nil. The Unit is reactive to crime reported by the public or submitted by divisional police officers**

Q5. What is the (a) name and (b) budget for each of the last 5 years for the cybercrime (or equivalent) unit?

A5. **HTCU Budget**
2010/11 -£59,000.00
2011/12- £59,000.00
2012/13- £65,153.00
2013/14- £65,153.00
2014/15- £65,153.00

Q6. Does the process for recording and passing cybercrimes to prosecutors differ from non-cybercrime?

A6. Crime recording is not a function of the HTCU.

Additionally, Gwent Police can neither confirm nor deny that it holds any other information relevant to your request by virtue of the following exemption:

Section 23 Information relating to the Security bodies;

Section 23 is a class based absolute exemption and there is no requirement to consider the public interest in this case.

Confirming or denying the existence of whether information is held would contravene the constrictions laid out with Section 23 of the Freedom of Information Act 2000 in that this stipulates a generic bar on disclosure of any information applied by, or concerning, certain Security Bodies.

Seized Computers

Added: 4 August 2014 Category: Information Technology

Reference: 17057

Disclosure

Q1. Does your force have specialists to analyse seized computers and if so, how many specialists do you have?

A1. Seven

Q2. If the computers are sent externally who are they sent to?

A2. None are sent externally.

Is a Computational Grid a Suitable Platform for High Performance Digital Forensics?

N. C. Pringle, (npringle@glam.ac.uk), I Sutherland (isutherl@glam.ac.uk)
Information Security Research Group
Faculty of Advanced Technology
University of Glamorgan
Pontypridd
CF37 1DL

Abstract: The size of computer storage media continues to increase at an exponential rate. In the summer of 2007, UK retailers are selling 500GB drives for £99 and manufacturers are starting to offer terabyte scale raid storage aimed at domestic market users who want to store and backup video, music and images. If we use current forensic tools on media of this size analysis time will increase to unacceptable levels; our analysis tools are only marginally more powerful than the application programs owned by the suspects. Most solutions under development offer speed improvements of, perhaps, ten times over current performance. To enable us to deliver prompt responses to investigations we need to improve the speed of the order of hundreds, if not, thousands of times greater than currently available. We have set ourselves the task of developing a high performance forensic system suitable for a regional crime facility. In this search for a practical solution we are assuming a budget of about UK £30,000. In this paper we review several alternatives and choose grid computing as the most promising. We offer a small grid primer and assess which parts of grid computing are, and are not, suitable for the processes in digital forensics. We assess developments when implementing grid systems in other disciplines and consider the advantages of adapting their technology and experiences to our needs. In answer to the important question "Is a grid system too bloated?" we offer our own benchmark testing.

1 INTRODUCTION

1.1 Size and quantity of media

In the last 30 years the increase in computing power available to 'ordinary' users has been quite incredible. In 1980 standard storage was a single 160k floppy disk. In 2007 manufacturers are starting to offer terabyte storage. While the capacity of the media has increased by about 5 million fold, the I/O rate has increased from .6MB/s to 300MB/s, an increase of 600 fold (Wikipedia 2008). This presents no problem to the end user, who is normally satisfied providing the applications run in an appropriate manner, but it does create a problem for forensic investigators in assessing and duplicating large volumes of data.

1.2 Timescales

It is accepted, as best practice, that we image the entire drive before analysis can start and then it is normal to run some analysis/indexing programs. From our own experience imaging a 500GB hard disk takes about 24 hours (at a typical speed of 10MB/s, depends on equipment, we can attain 24MB/s from specific equipment); the analysis of this data falls somewhere between two extremes. Sometimes we are asked to locate very specific information, perhaps an email sent at a specific time. In this case we can go straight to the PST file and search; the other extreme is a more complex search for interrelated information of evidential value. On 500GBs of media this is an awesome task. In these circumstances pre-processing is essential but this could take weeks. Forensic Tool Kit (Accessdata 2008) processes 1GB/hr on our slower equipment. Ironically, the more powerful computers become, the longer it will take us to access 'the whole of the media'.

1.3 Recent cases

Digital devices are surprisingly good sources of evidence. People often develop a very intimate relationship with their 'little toys' and mistakenly believe any secrets will remain so. Subsequently many people store information on their PCs, notebooks in particular, that they would not dream of committing to paper. Investigators are tending to turn to digital media as a *primary* source of evidence. During the summer of 2006 when there was the potential for a terrorist bombing campaign against transatlantic flights the security clampdown caused huge confusion for travellers and financial loss for the travel industry. During a televised press conference DCC Peter Clarke stated that the Metropolitan Police had made 27 arrests and subsequently seized 400 PCs, 200 mobile phones and 8,500 items of digital evidence (presumably CDs, DVDs, memory sticks etc) (BBC 2006). The disclosure went further to estimate this represented 6TB of data. If this estimate was true the PCs must have been fairly old and most of the extra media must have been floppy disks or perhaps CDs. If the PCs were on average only a couple of years old and had 40GB drives this would total, perhaps, 15TB of media. In a couple of years time there might be 100TB of media. In a verbal statement, broadcast on BBC radio on 15th June 2007 after the successful conviction of the terrorists, Peter Clarke praised the efforts of the digital investigative team saying that some officers, drawn in from the whole of the UK, spent the nights during the investigation in sleeping bags on the office floor such was the pressure of work to complete a substantial amount of the investigation with 14 days before the suspects had to be charged or released.

1.4 Big tasks – analysing video, data mining

Current forensic software is fairly basic. At most the software attempts to recover lost or deleted file and then organise these files into some classification to ease the manual search that follows; perhaps with regular expressions and some search software allows indexing and fuzzy logic and stemming, but we are some time away from semantic machine understanding. We can compare files by MD5 analysis but software that analyses images and recognises faces or places in some kind of automatic intelligent way is currently not available to the average investigator..

1.5 Establishing a processing gap between investigators and suspects.

Whatever the future presents we can be assured it will require more computer power than is available today. Unfortunately we cannot simply wait for more powerful PCs; as we acquire them so will the suspects. More powerful PCs mean more powerful capabilities; this will allow developers to create more rich experiences for the users. This will inevitably lead to yet more data. We are in a performance arms race. Digital forensics requires an advantage in terms of processing power over that of a potential suspects system. This all needs to be within a budget of a reasonable regional forensic facility. For this purpose we set an arbitrary figure of £35,000.

2 SIMILAR WORK

2.1 Breaking the Performance Wall

Surprisingly very little work has been published in the field of high performance forensic systems. It appears a single paper addresses the issue using distributed processing Roussev and Richard (2004) designed and built their own custom system stating that, in their opinion, existing distributed processing systems were too bloated to deliver the sort of performance they desired. Their system was intended to provide an investigator with the ability to make interactive searches of about 6GB of data with UNIX regular expressions (rather than pre-indexing). Their system passed files from a file server out to 8 hosts via a gigabit network. By placing the data in the RAM of each worker host they achieve significant improvements in performance. They chose not to use any higher-level protocols and wrote their system in C and TCP/IP bypassing facilities like SSH or RSH.

Even with the passing of just 3 years their target data set of 6GB seems rather trivial. We can now buy 16GB USB memory sticks! Their research was aimed at maximising pure interactive speed. We feel that this is desirable but we believe that sophisticated processing that would somehow rank the results to aid the investigator would be more useful.

Nonetheless we see the developments and subsequent experiments in their paper as a benchmark for distributed forensics systems. As we will see, in section 6, a grid system stands up well to the accusation of bloat made in this paper. Our own design and experiments with a grid performed well with the same architecture but improved significantly when we re-design the system.

3 SOLUTIONS WITHIN EXISTING SYSTEM ARCHITECTURE

3.1 Introduction

There are a number of ways we can improve performance on existing systems. Here we outline several and explain why they do not provide satisfactory solutions. As we assess them we must bear in mind we are looking for performance improvements of the order of hundreds or possibly thousands of times. When assessing their suitability we should remember our self-imposed budget of £35,000.

They are, in no particular order:

3.2 Multiple processors, multiple core processors and graphics processor units

We currently see hardware manufacturers designing with more powerful processors. Dual core and 64-bit technology are becoming standard as equipment on domestic machines. The cost of multiple processor systems grows exponentially with the number of processors [SGI 2008]. Systems like the SGI Altix 4700 system can have up to 512 sockets for dual core Itanium 64 bit processors that can address up to 128TB RAM, however they start at £8,000 for a dual processor blade module with 10GB RAM. Based on these figures it suggests a top specification system would be in the order of 256 times that price at about £2.5m which is out of our budget range. The use of multiple Graphic Processing Units has recently been explored (Marziale 2007).

3.3 Improved media I/O rates

Ideally we would simply increase I/O rates. Some kind of super Serial SCSI capable of Gigabytes of data transfer per second together with multi-core multi processor, but if sold commercially then this equipment could be acquired and used for illegal activity – furthering the performance arms race.

3.4 Increased numbers of investigators

The current, best, solution seems to be to have lots of PCs with lots of operators. This was the only solution available to the Metropolitan Police in 2006. People are flexible but are often the most expensive and increase the potential for human error so this cannot be classed as a *best* solution. This approach has problems of analysis quality control, as humans are liable to interpret information differently and considerable problems of co-ordination and cross-referencing.

3.5 Optimising existing hardware and operating systems

Working in our own forensic lab we found that storing the images and results on a RAID array attached to a central server and then processing the information on workstations connected by a Gigabit network was up to 10 times slower than local processing on the workstations. Moving the data closer to the processing i.e. on the local disk rather than a network shared drive on a server via a Gigabit network has a significant performance benefit. We lost some of the ability to group all of our images together and therefore process from *any* workstation, but gained a 10-fold speed improvement. This is an inexpensive, but limited, performance improvement.

3.6 Clever problem solving

There is always an argument for clever problem solving. These solutions will always improve performance because they can be applied to any system. Ultimately we would have to balance gains against development cost but in general these are not a problem to researchers. There are efforts at the present time to improve the performance of carving tools (DFRWS, 2006).

3.7 Beowulf clusters, MPI and PMV

Parallel processing on a system scale has been developing since the late 1950s mainly as a reaction to the costs of solutions considered in 3.2. Several techniques have been used starting with exotic hardware solutions, through to Messages Passing protocols of tightly clustered hosts. These systems are often used to deal with closely interrelated data, for example, flow modelling in fluid dynamics where fluid is divided into blocks, often called atoms that interact with those surrounding them. As the simulation moves forward many variables are passed between processes running on other hosts. But Digital forensic analysis presents a different problem, tasks are largely discrete. The creation of an MD5 hash on one file has no bearing on the same process on another file. The same is true for image analysis, text searching and other forensic processes. The parallel processing community call discrete distributed processing *embarrassingly easy parallelisation* as it does not require the programmer to deal with the complexities of synchronisation. One advantage of this solution is that it is very scalable. Beowulf clusters of many thousands of machines have been built, for example, the NCSA's Abe cluster of 1200 Dell PowerEdge 1955 hosts (Meneu et al, 2007). As they compose of conventional Mass Market Commodity PCs (M^{2}COT) costs are relatively low and they largely run common operating systems like Linux, as does Abe.

3.8 Grid systems

Ultimately unique, custom-made systems would provide the greatest performance as they could be specifically tailored to the task but the development cost of custom-built systems would surely be prohibitive in both time and money, and there would not be the benefit of the support of the wider development community. If we adopt a solution that comes, in at least part, from mainstream developments we can, perhaps, gain from developments in other fields. The current progress in the area of High Performance Computing is something that may benefit the forensic community.

Grid computing is the latest area of development in distributed HPC. Mainly driven by the needs of Physics, grid computing uses the standard technologies and protocols that drive the Internet to loosely couple huge numbers of PCs together in virtual organisations that span the entire world. The design tends to provide processing power to discrete tasks. Latency is a problem in these systems. Firstly a host willing and able to take the job has to be located, the data must then be available either by link or actually copying and then the results must be returned to the initiator or a location of their choosing. On small jobs there is a risk that the latency can exceed the job execution time. Because of this grids are often associated with tasks that require considerable processing time.

3.9 Conclusions

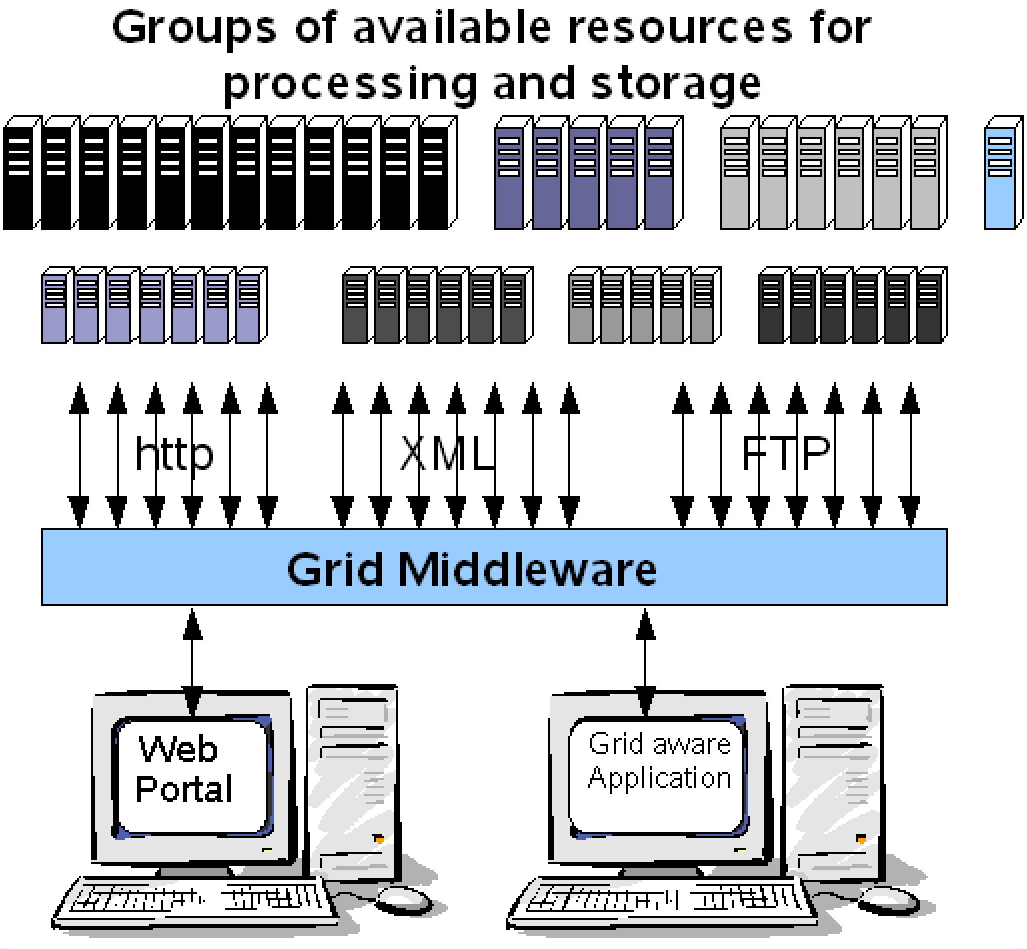
Alternatives such as the development of a super-chip or revolutionise, in secret, I/O speeds for memory devices for forensic purposes is unrealistic. It seems clear to us that adapting and adopting grid technology would be a good strategy benefiting from cross fertilisation of ideas and developments. Our conclusion is that distributed processing using grid architecture is the only viable way forward.

4 A TYPICAL GRID

We feel it would be useful at this stage to offer a very small and simple example of a grid computing.

4.1 Introduction

Grid computing has grown out of the drive for cheaper HPC over the last 20 years. It is built on the observation that the mass of processing power now exists in the unused clock cycles of ordinary PC rather than the mainframe environments. More often than not, ordinary PCs provide the storage and processing units linked by common protocols in a similar same way as the World Wide Web. Resources and users are linked by specific middleware tailored to the specific needs of the task.



A simplistic Grid structure

5 UNUSUAL CHARACTERISTICS OF FORENSIC DATA

Digital forensic has several unique features.

- We work on many files. 100,000 is typical on a single PC.
- They vary greatly in size from 1k to 2GB
- They have many formats. Fixed record, ASCII text, encoded, encrypted.
- We need meta-data about the file itself.
- We would like corroboration of results from multiple processing sources.
- Access Control restrictions to evidence at a highly granular level.

6 IS A GRID TOO SLOW FOR FORENSICS?

The major difference between processing on a single PC and a Grid is data transfer. It is important to establish whether the time spent transferring data is greater than the advantages of parallel processing. To determine if a grid-based system would be slower than Roussev & Richard's (Roussev 2004) custom system we attempted to run a series of comparative tasks on similar equipment running Globus 4.

6.1 Roussev & Richard's equipment setup

Roussev & Richard's (2004) test rig comprised of a Dual Processor Xeon 2.4GHz with a 500GB SCSI (Ultra 320?) RAID 5 Array to store the original image. A Gigabit Ethernet network connected this to 8 x 2.4GHz Pentium IV hosts with 1GB RAM in each. They did not use the local hard disk in the nodes choosing to store local files in RAM.

They used a 6GB NTFS image from a Windows 2000 machine with about 110,000 and 8,000 directories. Roussev & Richard (Roussev 2004) state that they ignored files over 20MB (to be more representative of a real world task) their experiment did not detail the exact number of files. Our sample suggests about 5000.

They performed 3 tasks

Task 1 - CACHE – the controller (the first of the 8 workstations) draws files, one at time, from the file server and then copies them back out in a round robin fashion to the other 7 nodes where they are stored in RAM. As they had 1GB of RAM available this suggests that they loaded the RAM until it was nearly full. This is effectively a sequential test to set a benchmark. The nodes wait for files to be sent to them, the controller does all the processing.

Task 2 - LOAD – the nodes, each independently, draw off a subset of files from the file server and stores them in its local RAM. This achieves the same result but divides the task between the nodes. In reality the nodes would need to know which files to grab and tell the other nodes that they have the file and not to both then grab it again; more management.

Task 3 - 'Viagra' search v[a-z]*i[a-z]*a[a-z]*g[a-z]*r[a-z]*a This is a simple benchmark task. There should be a near linear benefit from running this in parallel.

6.2 A comparable test rig

It was difficult to match the original equipment exactly but we replicated the experiment on 7 x Xeon 2.3GHz hosts each with 1GB RAM and an 80GB SATA 150 disks connected by Gigabit networking. We ran Debian Linux with Globus 4. We obtained a 20GB image of a Windows 95 FAT disk with about 40,000 files on it. Roussev & Richard's (2004) state that they only selected files less than 20MB and presumably filled the RAM on each node i.e. about 650MB. For us this meant between 758 and 978 files sent to on each machine.

6.3 Benchmark using NFS mount & copy

As a base we set up 6 of the hosts to share an NFS export on the 'File Server'. We used a simple Unix `cp` command to distribute the files sequentially by sending to host H001, then H002, then H003 etc. Having distributed the data we them used '`grep`' to search on the individual hosts. After the first search the disk cache will leave most, if not all, the recently searched data in RAM. We chose to distribute data totalling about 60% of the RAM installed in each host to utilise the caching feature.

Host	files	Total Size MB	NFS Copy Time secs	HD Search Time secs	RAM Search time secs
H001	859	660	39	16	2.6
H002	978	648	36	17	3.1
H003	826	669	37	17	3.3
H004	758	654	33	19	3.1
H005	822	670	36	17	2.3
H006	838	664	37	19	3.4
Total	5081	3965	218	105	17.8

When we analyse the results we can deduce 3 base figures for the activities in this test

NFS Copy time
3965 MB in 218 secs = 18 MB/s
HD Search time
3965 MB in 105 secs = 38 MB/Sec
RAM Search time
3965 MB in 17.8 secs = 222 MB /Sec

These results are in line with Roussev & Richard's (2004) results so we feel confident that we are working on similar equipment. This clearly demonstrates that if you divided the data out onto separate hosts and then searching in RAM it is about 8 times faster, as was reported by Roussev & Richard (2004).

6.4 Parallelising the task for a Grid

A system design within the constraints of one language or architecture does not usually translate to another without some modification. There are several stages to consider

- Setup
  - Data Distribution
  - Program Distribution
- Program Execution
- Data return
- Data Collation

To transfer the data we used one of the command line utilities included in Globus 4 - `globus-url-copy`. One of the limitations with this utility is the time the program takes to authenticate. This highlights a difference in implementation, rather than the option of scanning the source and sending each file separately invoking a separate authentication, in round robin fashion as Roussev & Richard (2004) did, it is better to scan through the source collecting source candidate file names and submit this list as a batch file to `globus-url-copy` using just one authentication. So the program structure is:

1. Mount the image
2. Use linux '`du`' to select candidate files
3. Compile 6 files containing source and destination file names for each host. Aim for 650MB per machine.
4. Run 6 instances of `globus-url-copy` with the files as a parameters
5. Use `globusrun-ws` to run the search program Job File for `globusrun-ws` in XML format

Results with sequential `gsiftp` copy, h001 then h002 then h003 and so on are

Host	files	Total Size MB	gsiftp Copy Time secs	HD Search Time secs	RAM Search time secs
H001	859	660	94	16	2.6
H002	978	648	114	17	3.1
H003	826	669	88	17	3.3
H004	758	654	90	19	3.1
H005	822	670	88	17	2.3
H006	838	664	97	19	3.4
Total	5081	3965	571	105	17.8

This is worse than our NFS benchmark. `Gsiftp` copy time has increased to 571 seconds from 218 with NFS.

Clearly we are paying the price for distributing the data. This is because we are failing to take the opportunity to run `globus-copy-url` in parallel.

6.5 Parallel distribution.

If we repeat Roussev & Richard's (2004) second test by initiating the copying and searching independently on each node we get noticeably different results. With the NFS copy benchmark, the 6 copies complete in 192 seconds; only slightly faster than if we had run the copying in sequence. If we repeat the `globus-copy-url` copying in parallel the whole copy completes in only fractionally more than the time taken to copy just 1 host in 119 seconds. Clearly the `Gsiftp` is more efficient than NFS when under load.

6.6 System analysis

For a distributed system to show advantage the penalties of file distribution must be outweighed by the advantages gained from parallel processing. As distribution requires the data to be read from the original disk, transferred across the network and stored on the remote host, either in RAM or on the local hard disk, the advantages cannot be expected from a trival operation like a one pass search.

6.7 Batch processing

Firstly we should be considering batch bulk processing of the files. If the target files were the subject of a number of operations such as all of the following then the data transfer would be worth the time and effort:

- MD5 calculations
- Statistical analysis for steganographic information
- Image analysis for facial recognition
- Data carving
- Semantic analysis
- Writing style analysis
- Digital photograph noise analysis.

6.8 Distributed evidence

In our tests we worked from an image of the original evidence because we intended to mimic the work undertaken by Roussev & Richard (2004). This effectively doubles the workload, as we would have had to read the original to create the image and the read the image to distribute the data. In a working system based on grid architecture, it would be more efficient to distribute the data as digital evidence packages across the grid straight from the original; with the data already in place the node would volunteer to process the files in its possession.

Having distributed the data and run the initial search we clearly saw subsequent work was many times faster. In our tests, once distributed and loaded into RAM, we were able to perform `grep` searches on 3.6GB of data in about 3 seconds. At times it was down to 1.6 seconds.

We have a very well known precedent for this technique; Google. Google achieve their spectacular response speed by distributing parts of the database, called shards, across the RAM of the 15,000 base model PCs that make up a Google cluster (Barroso 2003).

7 OUTLINE OF A DISTRIBUTED FORENSIC SYSTEM.

It is not unrealistic to consider in the near future that tasks faced by a forensic system may include the average home system with 2TB storage, Within our budget of £35,000 perhaps we could assemble a Grid of 400 hosts, with the current trend of decreasing cost perhaps less than £100 each in 2015. This system could have nearly a Picabyte of data storage and 10 Terabytes of RAM.

What would a grid forensic system look like?

In a physical sense it would probably be a series of rack mounted servers, either blade or small tower depending on our budget. The vast majority of these would be 'standard' build PCs although some may contain special hardware designed for specific tasks like cracking encryption. This systems would be accessed by a series of PCs running an end user program or a portal program in a web browser on either Windows or Linux.

Evidence would be introduced to the grid from a data acquisition node. This would be one of many PCs on the grid equipped with write blockers and PKI encryption. It would either read the original evidence, or a duplicate image, and extract and convert the files and sectors of the original evidence into a tagged evidence format. It would distribute them throughout the grid storage facility adding provenance data to the package header. Multiple copies would be stored across perhaps 4 or 5 nodes to enable data redundancy and corroboration of results. The presence of an evidence packages would be registered only with the local node that holds the data; this is a decentralised system. A standard sequence of tasks, MD5 hash generation and validation, thumbnail generation, text indexing etc would be run on each evidence package while cached in memory. The abundance of processing power could allow advanced semantic analysis extract meaning from the data. This software might come from work on the humanities grid. **REF to support this statement Photographic image analysis software could be used to match the contents of images, not in a simple pattern matching but by true image interpretation. This software might come from grids used in the film industry.*REF to support All these results would be stored in the evidence database on each node as part of a distributed database system.

Client analysis software would create XML job descriptions that ask for specific tasks to be carried out on specific files and send them to one of several job managers running on the grid. These would first query the evidence results database to see if the task had already been run. If so, it would immediately report back with these results, perhaps ultimately offering results of similar requests based on some kind of artificial intelligence analysis. If the job manager decides there are no useful results in the evidence database it would add the entry to a job database. Nodes could query the database for any task to be processed on data they possess. If they have the data they take the job. If not, they will take the job after a time-out when no other node has taken the job. The node may then have to acquire a copy of the data. . Ultimately the results would be added to the evidence database for future use. The data would be returned to the initiating client for human analysis.

Meanwhile a data-mining program would constantly sift through the database collating data with relationships that the investigator did not consider.

Grid computing is generating such great interest by such a huge and diverse community that if we too adopt grid technology for future large forensic systems it may be that we can benefit from the next generation of distributed software tools designed for other subjects.

8.0 SUMMARY AND CONCLUSIONS

In this paper we have explained that over the next few years the sheer quantity of data needing examination will present huge problems for the current generation of software forensic tools. There appears to be a lack of research in this area with the authors finding only one paper published on Distributed Forensic Systems (DFS).

After we considered the nature of forensic tasks, we assessed existing architectures and considered their appropriateness to solving the task of investigation. Grid computing was selected as a likely candidate because of low cost, favourable characteristics of scaling and suitability to the discreet nature of the forensic task.

Appreciating that Grids are not familiar territory for forensic investigators we then offered a Grid primer. We acknowledged some similar middle-ware and considered how middle-ware designed for forensic analysis might be different.

Having selected a most promising solution we addressed the issues raised in the only paper yet published addressing Distributed Forensic Systems and from our own testing concluded that the main accusation, of management bloat, seems unfounded. After redesigning the DFS to suit grid architecture we highlighted the key differences, namely distributed data. Finally we described a potential DFS.

REFERENCES

AccessData Forensic Toolkit
www.accessdata.com [last accessed 25 Jan 2008]

Barroso, L. A. (2003), Web Search of a Planet: The Google Cluster Architecture, IEEE Micro, March-April.2003

BBC News, Eleven charged over 'bomb plot',
http://news.bbc.co.uk/1/hi/uk/5271998.stm, [last accessed 13/12/2006]

Condor: High throughput computing., 2001,
http://www.cs.wisc.edu/condor/ , [last accessed 10/01/2008]

DFRWS, 2006, Digital Forensic Research Workshop 2006 Conference Challenge,
http://www.dfrws.org/2006/challenge/ [last accessed 12/01/2008]

Foster, I., Kesselman,. C., (1999), Computational Grids., Chapter 2 of "The Grid: Blueprint for a New Computing Infrastructure", Morgan-Kaufman.

Foster, Kishimoto, H. Savva, A. Berry, D. Djaoui, A. Grimshaw, A. Horn, B. Maciel, F. Siebenlist, F. Subramaniam, R. Treadwell, J. Von Reich J. (2005) The Open Grid Services Architecture, Version 1.0. I.. Informational Document, Global Grid Forum (GGF), January 29, 2005.

Foster. I (2006), Globus Toolkit Version 4: Software for Service-Oriented Systems. IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2006.

Historical Notes, about the Cost of Hard Drive Storage Space
http://www.littletechshoppe.com/ns1625/winchest.html [last accessed 25/01/2008]

Large Hadron Collider, 2008
http://lcg.web.cern.ch/LCG/ [last accessed 26/01/2008]

Marziale, L., Richard G. III, Roussev, V. 2007, Massive threading: Using GPUs to increase the performance of digital forensic tools, Digital Investigation, Volume 4, Supplement 1, September 2007, Pages 73-81

Meneu H., Strohmaier, E., Dongarra J., Horst, S., (2007), Top 500 Supercomputer Sites,
www.top500.org [last accessed 28/01/2008]

Nimrod/G (2000) Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid, 2000,

Open Grid Forum 2007,
http://www.ogf.org/About/abt_overview.php [last accessed 5/01/2008]

Roussev, V. and Richard, Dr G., 2004, Breaking the Performance Wall: The Case for Distributed Digital Forensics, Digital Forensics Research Workshop 2004, On-line at http://www.dfrws.org/2004/day2/Golden-Perfromance.pdf, accessed 14th September 2004

SAM-G,The SAM Grid Project 2007,
http://projects.fnal.gov/samgrid/, [last accessed 5/01/2008]

SGI, 2007,
www.sgi.com [last accessed 5/01/2008]

Sun 2007,
http://gridengine.sunsource.net/ [last accessed 5/01/2008]

Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, P. Vanderbilt, D. Snelling (2003) Open Grid Services Infrastructure (OGSI) Version 1.0. S.; Global Grid Forum Draft Recommendation, 27/6/2003.

Wikipedia – List of device bandwidths
http://en.wikipedia.org/wiki/List_of_device_bandwidths [last accessed January 2008]

Digital Investigation 11 (2014) S36–S44

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

Information assurance in a distributed forensic cluster

Nick Pringle[*], Mikhaila Burgess

University of South Wales (formerly University of Glamorgan), Treforest CF37 1DL, UK

Keywords:
Digital forensics
Distributed processing
Media analysis
FUSE file-systems
Information assurance

ABSTRACT

When digital forensics started in the mid-1980s most of the software used for analysis came from writing and debugging software. Amongst these tools was the UNIX utility ‘dd’ which was used to create an image of an entire storage device. In the next decade the practice of creating and using ‘an image’ became established as a fundamental base of what we call ‘sound forensic practice’. By virtue of its structure, every file within the media was an integrated part of the image and so we were assured that it was wholesome representation of the digital crime scene. In an age of terabyte media ‘the image’ is becoming increasingly cumbersome to process, simply because of its size. One solution to this lies in the use of distributed systems. However, the data assurance inherent in a single media image file is lost when data is stored in separate files distributed across a system. In this paper we assess current assurance practices and provide some solutions to the need to have assurance within a distributed system.

Introduction

The notion of using distributed processing to address the increasing scale of forensic investigations was first considered in “Breaking the performance Wall” in 2004 (Roussev and Richard, 2004). Despite being revisited several times since, (Ayers, 2009; Beebe, 2009; Garfinkel, 2010; Pringle and Sutherland, 2008; Richard and Roussev, 2006; Richard et al., 2007), this has not been developed and adopted as a workable solution. There has been a resistance to the idea of using an architecture where the data is moved and stored on a multitude of hosts for processing. In this paper we briefly consider the technical issues but conclude that the most important reason is the lack of a forensically sound approach to ensuring information assurance within a distributed system. This is required to ensure evidence management is regulated and clearly accountable for the legal community.

We will introduce our design for a middleware distributed processing solution, FCluster, which is specifically designed to provide assurance for the integrity of data.

Background

As digital forensic investigation methodologies have matured to accommodate the developments in technology, crime and investigative capabilities over the last 20 years, internal controls have been introduced to provide assurance standards required by the legal process.

Within our expectations of assurance there are a relatively small set of acceptable and ‘trusted’ investigative tools. FTK and EnCase are two of the most popular and trusted tools for digital media forensics. We know from more than a decade of use that their design endows confidence in the investigative process, and this is supported by these tools being tested for forensic appropriateness by NIST. In particular, the risk of ‘mixing up data’ between the evidence media and the host computer is negligible. There is no realistic way that data from another image could be introduced because there is no mechanism, other than operator error working on the wrong image, for this to

[*] Corresponding author. Tel.: +44 0 1443 4 83261.
E-mail address: nicholas.pringle@southwales.ac.uk (N. Pringle).

http://dx.doi.org/10.1016/j.diin.2014.03.005

happen. Provided the investigator is trained to use these applications as they were intended, the system is inherently assured. The designers consciously choose not to have a write-ability, not because it's just easier that way but because we have a special need to protect the data under investigation.

Assurance standards applicable to digital forensics

Unfortunately there are no explicit rules to define Information Assurance for processing Forensic data. Forensic evidence must adhere to the Daubert principle and the Federal Rules of Evidence in the US, ACPO guidelines in the UK (ACPO, 2012) and corresponding criteria elsewhere. ISO 27037 (ISO 27037:2012, 2012) addresses the acquisition and preservation of digital evidence but uses language such as "protected as far as possible" and that "evidence should be stored in an evidence facility that applies physical security controls". Standards like ISO17025:2005, intended for 'chemical' laboratories, have been the basis of digital forensic facilities but the translation from the analogue to the digital world is not always easy. ISO 27001:2013 defines characteristics of a management system that provides assurance, but not assurance itself. PCI-DSS (PCI Security Standards Council) does provide a more prescriptive standard but doesn't map well to digital forensics. When these are appropriate, unfortunately they are generally based upon the vague notion of 'best practice' and 'the accepted norm' in the particular field. It is difficult to apply in a rapidly developing domain, such as digital forensics, as technology changes are naturally always ahead of 'best practice' developments.

Internal controls in digital forensics

In practical terms, these reveal themselves in some of the characteristics of an existing system when, for example, a new item of evidence is introduced into the lab. It would first be recorded in some form of log. When the evidence image is copied onto the storage facility its success or failure needs to be validated, perhaps with a cryptographic hash digest, for example SHA-1, and this is recorded in the log book. The hash digest is an inherent property of the image. If the validation fails, the operator would investigate the process or equipment and make remedies and rerun the copy. This time, hopefully, it would succeed and the task is complete. Its success, and the previous failure, should both be recorded on the log book. In a paper system, the log book should have certain characteristics. The pages should be numbered and bound together. Anything written should be in ink. Lines on the page should either have writing or be lined through. If the log book is implemented on a computer system there should be an external verification, for example a time date stamp encrypted by PKI, that is beyond the capabilities of the operator to amend. These sorts of controls are common and should be familiar to any investigator.

All these processes should be subject to an *Audit*. By *Auditing*, we are checking that the system worked. The main problem with Auditing is that it is reflective and it often implies a protracted period of time passing before the audit. External audits are often annual, internal audits are perhaps, quarterly. It addresses issues that occurred in the past, assesses their conformance or non-conformance and should trigger changes in the system to prevent further breaches.

This was the case in the quality control employed in most industries in the Western World after the Second World War. Generally, goods were manufactured and were subject to quality control as a final stage where a sample set was tested for conformance. Those non-conforming were removed and either reworked or scrapped. The audit would trigger a period of reflection and perhaps modification to the production system to reduce the failure rate. Regrettably, there was an acceptance that a percentage of non-conformances would get through the system.

From audit to assurance

During the 1960s the Japanese introduced the idea of total quality assurance. The most important aspect of this was that controls were introduced before that action took place, not after.

The dictionary definitions give a sense of the retrospective nature of an audit (Dictionary.com, 2014) and the future intent of Assurance

Audit (*noun*)

1. **an official examination and verification of accounts and records, especially of financial accounts.**
2. **a report or statement reflecting an audit; a final statement of account.**

Assurance (*noun*)

1 **a positive declaration intended to give confidence; a promise**.

synonyms: word of honour, word, guarantee, promise, pledge, vow, avowal, oath, bond, affirmation, undertaking, commitment

2 **confidence or certainty in one's own abilities.**

synonyms: self-confidence, confidence, self-assurance, belief in oneself, faith in oneself, positiveness, assertiveness, self-possession,
self-reliance, nerve, poise, aplomb, presence of mind, phlegm, level-headedness, cool-headedness

Japanese production lines did not produce faulty goods because faulty components were not allowed to enter the production line. The effect of this change on the industrial base of the western world is a matter of history. During the 1970s and 1980s products from Japan surged leaving their North American and European competition behind, being viewed as unreliable. Modern management systems like Total Quality Management and Six-Sigma have their focus on controlling inputs and processes during the manufacturing process. Increases in quality, and customer satisfaction, are natural consequences of this approach.

Assurance in current computer systems

Most digital evidence from storage media presented in court is the result of analysis conducted using FTK or EnCase. This is so much a de-facto standard that we rarely question it but both systems are based on the same principles and on more than a decade of acceptance and precedence. At its heart is the idea of always presenting evidence originating 'from the image'.

Imaging tools usually make a cryptographic hash digest of either sections of data or the whole media. When the investigator copies the image onto the laboratory storage facility *they should* run a program to create a new cryptographic hash digest and compare it to the original to confirm the data is unchanged. There are a number of imaging programs used with varying assurance. The dd utility has no internal check-summing facility; both Expert Witness Format (EnCase) and Smart use file structures within their images to checksum every block, typically 64 KB. We are assured of the integrity of the data because it is seen as one complete, wholesome entity and is internally consistent.

It is largely left to the administrative system built around the computer system, as outlined in section 4, to provide assurance with existing tools that store or process these images.

Concepts like "Chain of Evidence" or Provenance have existed in legal proceedings for some time. Although users will take care of their data, the legal profession does pride itself on its highest possible standards in this matter. The ACPO guidelines describe this as a key task of the forensic practitioner.

Distribution of both data and processing

Current systems largely assume that the investigator will be handling a relatively small number of media items. In many investigations this might be only one or two forensic images. This is changing because case volumes are increasing (Justice FBI, 2012). To cope with this, it has been suggested that the next generation of forensic software could adopt a distributed processing model.

At this point we should make a distinction between distributed processing with centralised storage and distributed processing working with distributed storage. Having a distributed processing architecture that relies on a central, non-distributed, store of forensic images (Fig. 1) implies that the data has to be distributed to the processing nodes before it can be subjected to processing.

This is the case with FTK's 'distributed' processing. Processing time with this topology is dependent on the connection between the switch and the file server which rapidly becomes overloaded and limits scalability. We can mitigate this to some degree by building a storage facility based on fast SSD storage (450 MB/s), SATA III (600 MB/s) interfaces and even 10 Gb (1000 MB/s) Ethernet networking but this can be prohibitively expensive. Even this has limited capabilities in scaling out to even tens of processing hosts. Assuming we can make this investment it can still take many hours just to read the image off the storage media. If we wanted to conduct simultaneous analysis of several images held on the same storage facility it would have a significant impact on data dispersal time and so overall processing time.

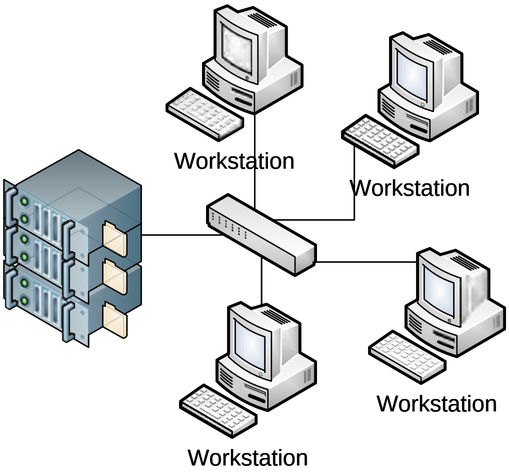


Fig. 1. The most common current architecture.

Digital forensics is not the only domain that has encountered this type of problem. Recently, Google solved their huge data problem by developing and applying a truly distributed data storage and processing model called Hadoop/MapReduce (Dean and Ghemawat, 2004). Although Hadoop/MapReduce provides distributed storage and processing it lacks the levels of assurance we require in processing data for presentation as evidence in legal proceedings. Distributed systems like Hadoop, Condor (Thain et al., 2005), Nimrod (Abramson et al., 1997), Weka (2014) and Globus (2014) have been slow to incorporate information assurance. Most have some access control but the users are more interested in getting their data processed than the nature of the environment in which it is stored. This is changing. Hadoop, for example, has been extended by commercial enterprises like Cloudera (2014), who realise that commercial acceptance now requires security but these are designed and built with general commercial markets in mind. In these systems it is often up to the user to exhibit diligence in the processing of a job. It is quite acceptable to not know where a file is stored or where it is processed; in fact it's a feature of Cloud Computing. There is some audit trail but rather like event manager in Windows, it is intended for performance and debugging issues rather than assurance. Information Assurance has greatly improved within Hadoop since Cloudera released CDH3 in April 2011 but it is unlikely that it will ever be extended to the exacting requirements of the legal process.

In all of these cases, information assurance has been added on as an after-thought. The assurance inherent in the legally established idiom of one investigator, one machine, one image would not be upheld in any current distributed storage/processing system. It's entirely understandable why software vendors don't support true distribution. It's doubtful that anybody would buy the product.

It would be much better if we could adopt a truly distributed storage and processing approach but built on a foundation of an assurance system rather than amend the existing systems.

Images, digital evidence bags and SIPs

The practice of acquiring a digital crime scene in the form of a 'forensic' image has served us well over the last 20 years but it is now under considerable pressure because of the size of media needing to be imaged. We feel this will lead to an increasing adoption of smaller units of storage. Well known formats such as AFF (AFFLIB) and DEB (Turner,

2006) have been established. We will refer to these collectively using a term taken from the ISO 14721, Open Archival Information System (OAIS) (OAIS, 2014), Submission Information Package (SIP).

Assurance in distributed storage

As soon as we split 'the image' into the SIPs that are needed to enable distributed processing we lose most of the inherent integrity of the 'oneness' of the image. Suddenly we have, perhaps, hundreds of thousands of SIPs to validate. In what way could assurance be re-established?

Information Assurance while processing SIPs could be provided in practice using some of the following methods:

- By a Property of an object: making and testing Checksums, Check digits, size, Control totals.
- By the Position/Location of the object: the fact that a file is in a certain location further enhances our faith that it is the correct one.
- By Loops of Authority and Acknowledgement: only accepting data from a device that was authorised to provide it.
- By Access control: allowing and denying.
- By Separation of process: having functionality provided by more than one program and clearly separating stages by function.
- By Audit trail: requiring independent sequential stamp, indelible records, recording with an authority.
- By Checklist: testing to see if previous checks have been completed and recording them in a table.

Introducing FCluster

FCluster is a middleware that provides an environment to allow forensic data processing to proceed with assurance. It is a means by which data integrity can be controlled; it is not an application program.

The design is based on the following assumptions, derived from current practice and technological developments.

- Media will continue to grow in capacity and quantity.
- That 'cross drive' forensics will be of increasing importance as individuals have many storage devices, crime is becoming more organised and forensic analysis systems will increasingly have to address multi agency interests.
- That multi-agency investigation will increase but will experience problems sharing and transferring large datasets.
- That, because of the above, the notion of the 'image' is becoming untenable but will be required for some time as a legacy. Instead evidential data will need to be stored as separate files across the system. Consequently we must expect tens, if not hundreds of millions of files in a system that stores the contents of many forensic images.
- That in most investigations, the evidence is found in 'the obvious place' and that most investigations are a case of locating and recording data found.
- The system should allow existing legacy software to run where possible. This new system should not require Guru programming skills with knowledge of devices like GPUs to gain access to huge processing power. However if such programs are developed it should allow these as well.
- That quantity of data requires the development of more automated tools. These can be simple reporting or correlating tools but need to run against large datasets.

The FCluster architecture

FCluster is a peer-to-peer middleware for a network of heterogeneous host computers. The prototype is built on Ubuntu Linux with future development planned for Windows and MacOS. Most of the code is either in C or Bash scripts. It uses MySQL, libcurl, ftp servers and ntfs-3g.

FCluster SIPs

FCluster includes a design for an SIP with a simple structure which only works with NTFS file-systems. This was done for the sake of simplicity when developing the prototype.

An FCluster SIP comprises of 2 parts (Fig. 2). An extensive header section contains XML delimited meta-data about the file's place on the original evidence media. This includes data from the file's entry in the NTFS $MFT and also a list of cluster numbers the file originally occupied on the source file-system together with an SHA1 for each of the clusters. The data section holds the file data which is encrypted using AES-256, with the key sent from FCluster, and then UUencoded to reduce problems in portability.

The SIPs themselves are named in a regular manner, [*VolumeID*]-[*SHA1*].meta. When the SIP is finally unpacked, decoded and decrypted on the FCluster the resulting file must have the same SHA1 as its filename suggests and is included within the header section of the SIP. To achieve this it must have been generated on the imaging device authorised by the key created by the FCluster when it authorizes imaging (see section 14.1) or it will not decrypt when it is ingested into the FCluster file system. These form two assurances, one of a property of the file, the name and the 'double entry' of the success of the encryption/decryption key.

FCluster subsystems

Applying a principle of separating processes, FCluster comprises of 4 sub-systems (Fig. 3).

These are explained in more detail in section 14.

```
      <metadata ntfs>
        This contains detailed information about the
directory entry and details of which clusters the file
occupied, together with SHA1 checksums for each
cluster
      </metadata>
      <data>
      This contains UUencoded version of the data
that was encrypted with AES-256 with the
acquisition authority key
      </data>
```

Fig. 2. FCluster SIP structure.

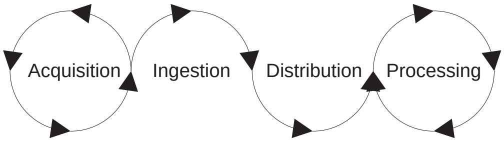


Fig. 3. The 4 subsystems with FCluster.

FCluster roles

Each host in the cluster might fulfil all the FCluster functions listed below, but it is likely that most hosts will be allocated just three or four roles.

- **Acquisition Authority**

That creates the cryptographic keys used to authorise imaging.

- **Imaging**

That creates the directory metadata SIPs, file data SIPs and Image files.

- **FClusterfs file-system metadata storage**

The heart of FCluster is a multi-featured File System in User Space (FUSE) file system (Filesystem using MyS, 2013) based around an SQL database.

- **SIP Ingestor**

That locates expected new evidence SIPs and triggers ingestion.

- **Load Balancer**

That chooses which storage/processing host should hold the primary copy of the data based on its workload.

- **Replicator**

That makes sure there are enough copies of the SIPs to ensure redundancy and also verification that the data is still valid.

- **Data Storage server**

That actually holds the data.

- **Processing**

Which actually does the processing. Almost always combined with the storage role.

We believe our proposed system provides assurance at every stage in such a way that the next stage cannot commence if the previous assurance is not satisfied. The core of this assurance is embodied in the use of FUSE file-system specifically designed for our purpose.

FClusterfs

We have observed that Map/Reduce implements a new file system, HDFS, as a base for its processing. This has been implemented as a middleware on top of the native file-system used by the operating system. We follow the same approach.

The use of FUSE to build custom file systems has been proposed within the digital forensics domain (Richard et al., 2007). FClusterfs advances the notion and uses the technique to provide a solution that addresses the key issues of Assurance in a distributed processing environment. It merges together several existing FUSE file systems to form a new file system.

FClusterfs is based on MySQLfs (Filesystem using MyS, 2013). MySQLfs employs an SQL database consisting of 3 tables to completely replace the native file system. The ‘inodes’ table provides storage for file metadata like names, dates/times, size, access rights etc usually seen as a ‘directory’. The ‘tree’ table stores the hierarchical structure of folders and filenames found in the file-system. The 3rd table ‘data_blocks’ stores the actual data as a series of binary large objects (BLOBs) replacing the clusters of the disk format.

In FClusterfs we use the tree and inodes tables found in MySQLfs. FClusterfs provides read-only access and so we never need to manipulate directories. We have a table called ‘meta-data’ to store the meta-data from the original location of the data. This is a variable length, large text field and so is better in a table of its own.

A single FClusterfs database can store many file-systems. We have a table, VolumeInformation, which contains a record of each file-system stored within the inodes table. We have added a field ‘VolumeID’ to inodes to identify which file-system the entry relates to.

We substitute the functionality of the ‘data_blocks’ table in MySQLfs with the ability to read data stored on remote servers. We have chosen to connect to the remote servers using the ftp protocol because of the features of another existing FUSE file-system curlFTPfs (Robso, 2013). curlFTPfs allows the user to mount a connection to an ftp server and make it appear to be part of the host’s file system. curlFTPfs attains much of its power and flexibility because it is based in the libcurl library and can support not only ftp but SSH, SFTP, HTTP, HTTPS but, despite known security issues with unencrypted data transfer, we have chosen ftp as a simple base for a prototype. In a real world scenario, SSH would be a more robust protocol. curlFTPfs only allows one ftp server per mounted file system. In FClusterfs we have enhanced this to be able to access individual files on any ftp server on a file by file basis. The corresponding server details are stored in the file’s record in additional filds we have added to the ‘inodes’ table. When the user sees a directory listing in their user space it appears as a continuous list drawn from the ‘inodes’ table but in reality each file’s data will be on a ftp server which is most likely remote. Each file is held in its entirety on the ftp server. In the prototype the entire file is transferred and held in cache in memory. In curlFTPfs 128 MB chunks are transferred just once and, if the file is over 128 MB, a mosaic is built in a cache in local memory.

It is important to realise that although FClusterfs does allow data to be transported across the Ethernet network, it

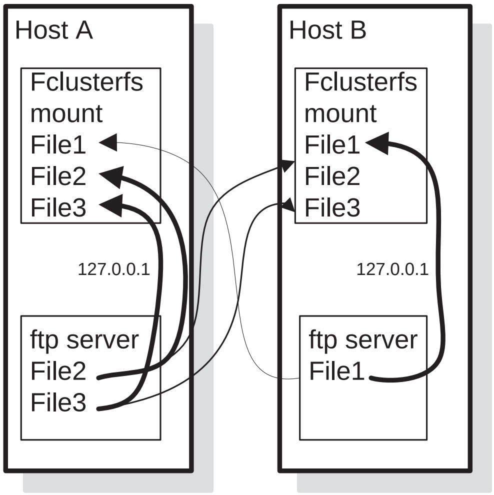


Fig. 4. ftp via 127.0.0.1 localhost loopback.

is primarily a means of standardising access to data held locally on its own ftp server. When we use the term 'passes across the network' it should be taken as via 127.0.0.1, the localhost loopback connection (Fig. 4).

FCluster is also peer to peer and so any node can mount a directory that can reference files on any server (Fig. 5).

Further, data held on this multitude of ftp servers is encrypted and uses techniques from ecryptfs (Hicks et al., 2013) to decrypt data on-the-fly. After it leaves the ftp server media, it passes across the network and is decrypted in the user's host before being held in cached space in RAM in their Virtual File System.

As previously mentioned, FClusterfs is read-only. There is no code to provide functions like write/delete/chown/chmod. This is a fundamental requirement of a forensic system and, fortuitously, greatly simplifies the code.

FCluster has auditing which it draws from Loggedfs (Flament, 2013). Loggedfs' audit is felt to be too granular for our purposes and instead we choose to record only significant actions like SIP movement, unpacking and the opening of data-files for processing. Recording access to parts of a file is felt to be unnecessary and would only slow the system and make the logs unreadable. All audit records are stored in a table 'audit' recording date/times, users.

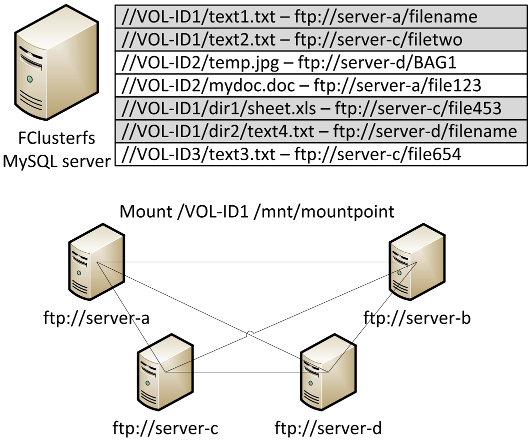

//VOL-ID1/text1.txt – ftp://server-a/filename
//VOL-ID1/text2.txt – ftp://server-c/filetwo
//VOL-ID2/temp.jpg – ftp://server-d/BAG1
//VOL-ID2/mydoc.doc – ftp://server-a/file123
//VOL-ID1/dir1/sheet.xls – ftp://server-c/file453
//VOL-ID1/dir2/text4.txt – ftp://server-d/filename
//VOL-ID3/text3.txt – ftp://server-c/file654

Fig. 5. FCluster mounts peer to peer.

Although the data location url information is available to the user eg ftp://myserver.com/, the username and password needed to login to the ftp server and gain access the data is not. It is held in another table 'serveraccessinfo' and is retrieved on-the-fly during a read request by FClusterfs. Users can only access evidence via the FClusterfs file-system which provides data from the ftp servers.

FClusterfs is intended to completely replace any need for network shares like NFS or SMB but to emphasise, although FCluster provides access to the file-system under investigation and will work over a network connection it is not the most effective way to work. It is intended to process local data by the host of the ftp server holding each of the files. The location, url, of the ftp server hosting the data is part of the 'inodes' table extending the fields used by FClusterfs and so the 'locality' of the file can trigger the processing task to be initiated within the host.

Mounting the FCluster file-system

The behaviour of an FClusterfs file system is defined when it is mounted by a command line which contains the following entries:

```
fclusterfs
–mysql_user=me
–mysql_password=mypassword
–mysql_host=25.63.133.244
–mysql_database=fclusterfs
–volume=74a8f0f627cc0dc6
–audituser='Investigator Name'
/home/user/Desktop/fsmount
```

Multiple file systems can be mounted on the user's host system and multiple SQL servers can provide storage for FClusterfs file-system databases.

Functional overview – dataflow

Having established the component parts of FCluster we can now demonstrate its operation by following data as it is gathered and passed into the system.

The initial imaging process has three deliverables:

1 a SIP containing directory metadata.
2 a collection of SIPs, one each for each file that falls into a 'high value' criteria set by the image acquirer.
3 a conventional 'forensic image', for reference and later extraction of further data.

The selection of files to be packaged as SIPs takes a prioritised triage approach collecting only file types expected to have a higher likelihood of containing evidence depending on the case type.

The first stage of ingestion into FCluster is when the SIP, containing the data defining the file system directory, is imported into the MySQL database at the heart of FClusterfs. At this stage a directory skeleton will exist but no data is available within FCluster.

The file data, in the form of a number of SIPs, is imported as it becomes available. This starts a process of ‘filling out’ the evidence file system with data associated with each directory entry. The data is distributed across the Datanodes according to a load balancing algorithm which bases its allocation on benchmarking previously created by running a known set of approved programs against typical data files.

When a SIP arrives on its storage host, it is unpacked and its contents are verified in a number of ways. Only if it is proven to be valid is it then accepted and made available via the distributed file system, FClusterfs. Upon approval at its storage location, a defined list of tasks is invoked and automatic process is conducted, for example generating text indexing or thumb-nailing images.

To provide redundancy and secondary load balancing, a replication agent firstly ensures constant and routine validation of data by applying an SHA 1 checksum to each file; it can then ensure that there are multiple copies of the data, normally three, held on separate hosts within the cluster.

The SIPs at image time will have, most likely, captured only part of the evidence. Subsequently a ‘Bag it on demand’ system can trigger an on-the-fly acquisition of data that was initially deemed of secondary interest within the image once it has been completed and is available to the cluster. This data is validated and placed in the same assured manner as the rest of the system.

How FCluster is configured as a network system is up to the administrator but it can form a local or wide area network. The prototype successfully uses a VPN to connect the nodes. We’ve extended it to use nodes on Amazon Web Services. Whenever data is transferred between nodes is it always in an encrypted form and so can be considered safe in a technical sense but this may not be acceptable on principle within a legal environment. The primary objective, and the core of any speed improvement, is that processing takes place locally on the datanode holding the data. In a similar way to the use of SHA1s to identify ‘Bad’ files, the system can be used without the actual files being accessed. Results are transferred across the network but not normally the data.

FCluster by stages of assurance

FCluster has 4 ‘zones’ of assurance as shown in Fig. 7. We now step through then in more detail.

Acquisition assurance

The first assurance in the system is one of the “Loops of Authority and Acknowledgement” type in which authority is granted to an imaging device to take an image and then FCluster only accepts data that was gathered with that authority Fig. 6.

The FCluster administrator generates an ‘Authority to image’ in the form of a file which will be issued to a specific device. This file contains a reference number and a randomly generated key which will be used at acquisition time to encrypt the data stored in the SIPs. The reference

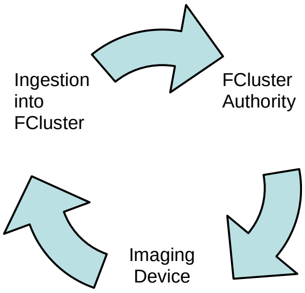


Fig. 6. Acquisition assurance.

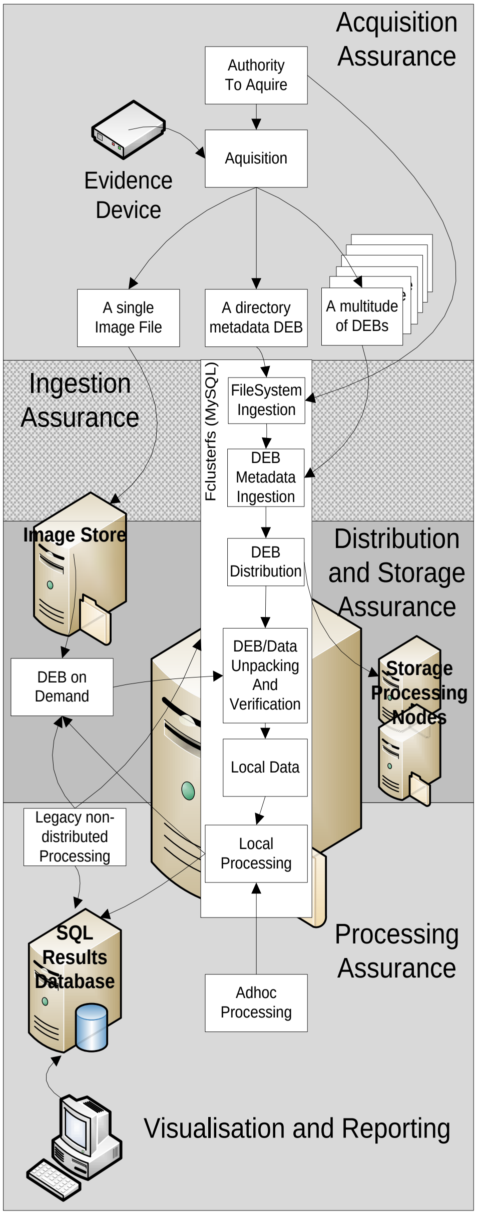


Fig. 7. The 4 zones of assurance.

number and key are recorded in the VolumeInformation table in FClusterfs. Multiple keys can be created and issued to multiple imaging devices to form a 'stock of authorities' to be used over a period of time, the keys have an 'expiry date' associated with them as an added control.

As previously explained, in section 13, the imaging process has three outputs. The SIP containing file-system metadata is the first to be imported. The reference number under which the cryptographic key was recorded is located in VolumeInformation table. If it is present, has not expired or has not been previously fulfilled, the import can proceed. The contents of the SIP is read and the directory meta-data is decrypted and records for each file are created in the inodes, tree and metadata tables. These include fields that describe the full path and filename, file size, MAC dates and times etc. If the key is not present in the VolumeID table, the import cannot proceed.

At the end of this process a complete 'framework' of the directory structure and filenames will have been created in the FClusterfs database. It is actually possible to mount this FClusterfs structure and traverse the directory but as the import of file SIPs that contain file data has not been carried out there is no actual data to analyse in the files.

Ingestion assurance

We now use a series of "checklists" to control the import of the details and contents of the data-file SIPs.

The SIP staging directory, where SIPs are placed ready to be imported, is scanned and any SIPs which form part of a Volume that is expected to be imported are found and the header is read to extract details of the VolumeID, path, filename and size. The inodes table of FClusterfs is searched to see if this SIP is expected, ie there is an entry previously made by a file-system SIP import. At this stage, various fields like the original file's SHA1 and staging directory url should be empty. If there is a record that satisfies these criteria then the fields in the inodes table are populated with the meta-data extracted from the data-SIP. If there is a record in the inodes table and it shows it has already been imported it will not be considered again.

Distribution assurance

This stage has three components. Load Balancing, Moving SIPs to their primary destination and unpacking them.

Load balancing

Having ingested the volume directory metadata the system is now primed to expect the SIPs of data that makeup that file system. The selection of the primary storage of the data is the first task of the ***loadbalancer***. It allocates a storage server to hold the data held within the SIP and records this in the FCluster inodes table. Allocation is based on the available capacity of the host, its processing power and its estimated time to finish its current task list.

The movefile daemon

The movefile daemon also uses "checklist" type assurance by constantly scanning the inodes table of FClusterfs for any SIP that has been allocated a datanode, not been marked as being 'in place' and where the evidence SIP is staged in a local directory. If these conditions are met the SIP is transferred to the storage datanode as allocated by the loadbalancer. If, and only if, the transfer is successful does movedata update the inode table with 'primarystorageinplace' set to true. Movedata is the only mechanism whereby actual data can be moved around the system. It can only operate when all the preconditions from Ingestion Assurance are met. It does not simply scan an evidence folder and move whatever SIPs are present; it moves only expected SIPs, as recorded in the FCluster inodes table, from a folder.

The unpack daemon

Unpacker daemon constantly scans the inodes table to see if there are any SIPs that are on their local server but not unpacked. It takes the entry from the database and looks to see if the files are on its ftp host, as should be the case from the entries in inodes, not the other way round. A file that simply arrives on the server without an entry in inodes would be ignored. When a suitable SIP is identified it is split into header and data sections. The header, containing the metadata is inserted into the 'meta_data' table and the header file erased. The data section is undecoded and the data decrypted with a key stored in the VolumeListing table. This was the key first created and issued by the FCluster and used to encrypt the data in the SIP at acquisition time. If the key does not work, the file cannot be decrypted and so unpacking would fail. Only if the file decrypts and the resulting file has an SHA1 checksum that matches both the name of the file itself and the SHA1 as recorded in the inodes table is the datafile finally accepted.

Processing assurance

The task daemon scans the tasks table to see if any job is required for a file that it holds locally. Because all file access must take place by utilising the enhanced FClusterfs file-system the file must be the correct file and must have the original content that was collected at imaging time. FClusterfs also gives us fine grained access control to the files within a file system. We could, if we wished, control which users can process specific data with specific programs.

Conclusions

We have demonstrated that by ensuring a rigorous protocol when importing SIPs into a distributed cluster we can provide a level of assurance in data transfer and storage. Additionally, by adopting the same approach as Hadoop we have created a prototype of a middleware specifically designed to address the assurance requirements required in the legal process while providing effective distributed processing. As to whether this does achieve an acceptable level we offer this design for further debate. It should be clear that this design draws upon knowledge from many domains and so there is no single criteria set that can be applied.

Speed concerns

A primary concern with FClusterfs is speed but in practice this has not proven to be a significant problem. Firstly, file access in existing systems is often across a network connection via SMB and NFS shares. FCluster does this in the same way but using the ftp protocol. These are roughly equal or perhaps slightly slower. Secondly, as we have made clear, FCluster is read-only and so has no record or file locking code. As a result, even when FCluster draws from a remote ftp server data is cached locally in RAM and never needs to refer to the source for updates or changes. Thirdly, the system is designed so that each storage host should process its own local data, so the network issue completely disappears.

All distributed systems suffer from a management overhead. This management issue exists in single host solutions but is exacerbated when management data has to be passed in messages across relatively slow network connections rather than using local memory. This limits scalability but in our initial test we find that the effectiveness of clusters of about 50 hosts on a local Gigabit network does not degrade significantly.

As of Spring 2014, the FCluster prototype is almost complete and we are starting full assessment. We intend this to be available when complete via www.fcluster.org.uk.

Future work

There are many areas in this design that present the opportunity for further research. Our own priorities would include rearranging the database structures to implement the principle of division into subsystems so that the assurance subsystems are reflected in the arrangement of data within the tables. On network security the use of ftp, only used as a protocol for ease when building a prototype, should be replaced with, for example, SSH and use digital certificates as authentication. Issues of Governance and Chain of Custody need to be assessed including comparisons with standards like ISO 27037 and OAIS. The design was always intended to allow existing, legacy, software to run without alteration. We need to consider how we can achieve data abstraction above the middleware.

Acknowledgements

This work is part-funded by the European Social Fund (ESF) through the European Union's Convergence programme administered by the Welsh Government.

ESF

Llywodraeth Cymru
Welsh Government

Ewrop & Chymru: Buddsoddi yn eich dyfodol
Cronfa Gymdeithasol Ewrop

Europe & Wales: Investing in your future
European Social Fund

kess
Ysgoloriaethau Sgiliau Economi Gwybodaeth
Knowledge Economy Skills Scholarships

References

Abramson D, Foster I, Giddy J, Lewis A, Sosic R, Sutherst R, et al. The Nimrod computational workbench: a case study in desktop metacomputing. Australian Computer Science Communications 1997;19:17–26.

ACPO. ACPO good practice guide for digital evidence http://library.npia.police.uk/docs/acpo/digital-evidence-2012.pdf; 2012.

AFFLIB – the advanced forensic format. http://afflib.sourceforge.net/.

Ayers D. A second generation computer forensic analysis system. Digital Investigation 2009;6:S34–42.

Beebe N. Digital forensic research: the good, the bad and the unaddressed. Advances in Digital Forensics 2009;V:17–36. Springer.

Cloudera. http://cloudera.com/; 2014.

Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. In: OSDI '04: 6th symposium on operating systems design and implementation. USENIX Association; 2004.

Dictionary.com. http://dictionary.reference.com; 2014.

FUSE Filesystem using MySQL as storage.http://mysqlfs.sourceforge.net/; 2013.

Flament R. LoggedFS – filesystem monitoring with Fuse http://loggedfs.sourceforge.net/; 2013.

Garfinkel S. Digital forensics research: the next 10 years. Digital Investigation 2010;7:S64–73.

Globus. https://www.globus.org/; 2014.

Hicks T, Kirkland D, Halcrow M. eCryptfs, a cryptographic stacked filesystem for Linux http://ecryptfs.org/; 2013.

ISO 17025:2005. General requirements for the competence of testing and calibration laboratories. ISO; 2005.

ISO 27001:2013. Information technology – security techniques – information security management systems – requirements. ISO; 2013.

ISO 27037:2012. Information technology — security techniques — guidelines for identification, collection, acquisition, and preservation of digital evidence. ISO; 2012.

Justice FBI, U.D. of. Regional computer forensics laboratory annual report for fiscal year 2012 http://www.rcfl.gov/downloads/documents/RCFL_Nat_Annual12pdf; 2012.

OAIS. http://public.ccsds.org/publications/archive/650x0m2.pdf; 2014.

PCI Security Standards Council. https://www.pcisecuritystandards.org/index.php.

Pringle N, Sutherland I. Is a grid a suitable platform for high performance digital forensics?. In: The 7th European conference on information warfare and security; 2008.

Richard III G, Roussev V. Digital forensics tools: the next generation. Digital Crime and Forensic Science in Cyberspace; 2006:75–90. Idea Group Publishing.

Richard III G, Roussev V, Marziale L. Forensic discovery auditing of digital evidence containers. Digital Investigation 2007;4:88–97.

Robson BA. CurlFtpFS – an FTP filesystem based on Curl and FUSE http://curlftpfs.sourceforge.net/; 2013.

Roussev V, Richard III G. Breaking the performance wall: the case for distributed digital forensics. In: Proceedings of the 2004 digital forensics research workshop; 2004.

Thain D, Tannenbaum T, Livny M. Distributed computing in practice: the condor experience. Concurrency and Computation: Practice and Experience 2005;17:323–56.

Turner P. Selective and intelligent imaging using digital evidence bags. Digital Investigation 2006;3:59–64.

Weka. Waikato environment for knowledge analysis http://www.cs.waikato.ac.nz/ml/index.html; 2014.