AlmaNebula: a computer forensics framework for the Cloud

Corrado Federici*

University of Bologna, Italy

Abstract

Scalability, fault tolerance and collaborative processing across possibly dispersed sites are key enablers of modern computer forensics applications, that must be able to elastically accommodate all kinds of digital investigations, without wasting resources or fail to deliver timely outcomes. Traditional tools running in a standalone or client-server setups may fall short when handling the multi terabyte scale of a case just above average or, conversely, lie mainly underutilized when dealing with few digital evidences. A new category of applications that leverage the opportunities offered by modern Cloud Computing (CC) platforms, where scalable computational power and storage capacity can be engaged and decommissioned on demand, allow one to conveniently master huge amounts of information that otherwise could be impossible to wield. This paper discusses the design goals, technical requirements and architecture of AlmaNebula, a conceptual framework for the analysis of digital evidences built on top of a Cloud infrastructure, which aims to embody the concept of "Forensics as a service".

1. Introduction

Modern Forensic Computing, the science that deals with techniques and procedures for identifying, collecting, preserving, analyzing and presenting digital data before a court of law [1], requires a sharply increasing amount of computational resources as the number of computer related investigations continues to grow. An efficient strategy that keeps acceptable delivery times calls for gigantic storage areas and huge computational power, not only to visualize content from a single device, but also to extract actionable information from a collection of evidences analyzed as a whole. In this scenario, forensic tools based on a traditional standalone or three tiered approach (client, application server and database) may fall short because of their intrinsic inability to scale. Classic architectures resort to over provisioning to accommodate demand bursts but, due to the wide difference between peak and average utilization [2], their resources may lie undersubscribed. Conversely, facing the "Big Data" issues forced cloud pioneers

*E-mail address: corrado.federici@unibo.it
like Amazon or Google to devise intelligent distributed file systems running on cheap commodity servers that can process concurrently chunks of data before consolidating the result. Digital Forensics requires a degree of processing power on large collections of documents which has much in common with the Big Data handling that can offer its technological advances to evidence analysis.

Contribution: this work delves into a set of design principles, technical specifications and conceptual architecture of a forensic platform that leverage the power and storage capacity of private/community Clouds. A modular petabyte-scalable infrastructure geared towards the automatic extraction of actionable knowledge from a collection of digital evidences exposed by means of intuitive interfaces. This aims to embody the concept of "Forensics as a service", a facility for examiners with very basic technical experience that public or private organizations may utilize to grasp all the benefits offered by the utility computing paradigm.

Roadmap: the rest of the paper is organized as follows: next section surveys related work. Section 3 gives some background information and discusses the limitations of current approaches. Section 4 presents AlmaNebula's design goals and requirements, while section 5 covers its architecture. Conclusions are drawn in section 6.

2. Related work

Papers on advances of forensic platforms [3] stressed the need of a new class of applications that could harness the power of distributed computing as standalone forensic tools, albeit well designed, could fail to deliver timely results. In [4] authors acknowledged that MapReduce programming model (MR) [5] is a powerful conceptual model for describing typical forensic processing and devised a framework, named MMR, based on the Phoenix [6] shared memory implementation of Map Reduce. The Sleuth Kit Hadoop Framework [7] is a very interesting experimental project that relies on Hadoop to build a distributed system for evidence content extraction, analysis and reporting that is amenable for a cloud deployment. Despite many useful analysis features like text extraction, keyword search and document clustering have been implemented, there is no user interface yet and process outcomes are delivered as JSON report files. The Open Computer Forensic Architecture (OCFA) [8] is a well designed forensic platform organized in pluggable modules which aims to automate content extraction from a large amount of digital material and creates a searchable index of text and metadata. However, module development follows a proprietary schema and persistency of data is delegated to a sound, but monolithic PostgreSQL database. In [9] an outlook of the digital forensics research in the next 10 years is presented, where the author reviews the limitations of today's tools and finds that they are monolithic applications designed to make visible what investigators are looking for, when the mere presence of a file is an evidence of a crime, but fail to detect information that is out of the ordinary or out of place. The need of more intuitive user interfaces able to present information and knowledge to analysts and not only mere data is also covered in [10].

3. Background

Computer Forensics teams, which typically run understaffed, would appreciate the opportunity to be relieved from daily IT management activity and, more importantly, exploit the potentially vast computational power and storage capacity of Cloud Computing (CC) to harbor and analyze digital data. The Cloud would make it possible to create elastic forensic analysis platforms able to cope with demand peaks with no service disruption and, conversely, no fear of resource wastage during idle times. Theoretically, it could be admissible to host digital evidences in a public Cloud, if the provider were able to offer Government certified services with proper security category. In this respect, there are outstanding examples of risk management programs for the Public Sector, such as the United States FEDRAMP [11]. Nevertheless, sharing control on high sensitive data may raise concerns about their confidentiality and compliance to norms and regulations. Therefore, the natural conclusion is that, at this stage of maturity of
public cloud offers, a framework for evidence analysis is more likely to be targeted towards a private or community cloud deployment. This paper will not delve into legal implications and assumes that such a platform is always feasible as, at least in a private deployment with augmented security measures due to resource pooling, court authorizations that were received for evidence handling with traditional tools continue to stand.

3.1. Current approaches and limitations

The main freely downloadable solutions such as Autopsy or Dff, have important conceptual mainstays such as the modular architecture or the possibility to use a database as a central storage for data interchange among disparate modules. However, despite the great added value that these tools bring to the computer forensic community, it is worth noticing the following circumstances:

- **User interfaces made for experts**: all the aforementioned tools have user interfaces to allow operations that are close to the physical nature of devices and therefore feature file system browsing facilities that visualize the directory structure of disk partitions, with advanced capabilities such as enumeration of unallocated sectors or display of file raw content. Nevertheless, this wealth of details is not well suited for analysts, whose aim is to uncover logically hidden information buried in a huge mess of irrelevant data by exploiting their deep acquaintance to the case. Evidence-oriented design of interfaces [9] enable technicians to visualize what they are looking for, but do not much help investigators to extract and consolidate actionable knowledge.

- **Hardware platform scaling**: wherever possible, standalone platforms can temporarily improve their performance with vertical scaling, that is empowering the existing hardware by adding more CPUs, disks and memory banks. This is a rigid and coarse grained method to scale out, so there is some risk of average underutilization during periods of reduced demand.

- **RDBMS issues**: relational databases (RDBMS) are rock solid data storages which are necessary in all class of real time applications, like airline booking or e-commerce, which cannot tolerate an inconsistent database status that could be originated by concurrent user write access. Conversely, there might be some issues of performance and scalability with distributed RDBMS when the amount of handled data reaches the *Web scale*: inherently slow table joins or resource locking during transactions [12] just to name a few.

- **Basic security**: All the listed forensic tools rely on the authentication services either provided by the operative system or by the application itself. In a cloud scenario however, where value concentration may increase the attack surface, there is the need to strengthen the protection perimeter of information and single factor authentication schemes may no longer suffice.

4. Design goals and requirements

4.1. Design goals

Based on the previous assumptions, AlmaNebula's design rests on the following principles:

- **COTS driven scalability**: a major design goal is achieving an horizontal massive scalability by leveraging commodity off the shelf (COTS) hardware: no special shared redundant storage is requested, but directly-attached hard disks that every server can host internally. Overall capacity increase must be reached by seamlessly adding new computational units (nodes) to the network, with no theoretical upper bounds and without service disruption. As far as possible, nodes must be peer, without any specialized role that could become a single point of failure. Elasticity should be possible by mean of automatic facilities that keep under measure machine resources and decide autonomously to intervene when load reaches some upper or lower thresholds. Usage of COTS coupled with an high level of automation will contribute to lower maintenance costs and achieve a relevant degree of investment protection by leveraging existing hardware assets.
Resiliency and distribution: the platform must be able to tolerate faults by putting intelligence in the software layer and account for failures that would not be otherwise handled with COTS. A geographical distribution of the computational units must be possible and data replication protocols must be efficient and resilient enough to cope with temporarily slow or intermittent WAN links. According to a publicly available implementation of MR, rapidity of overall tasks execution will be achieved by splitting digital evidences into atomic entities to be distributed among all online nodes.

Loose ACID compliance: a strict RDBMS ACID compliance seems not so necessary for forensic applications, most of which are organized in a preliminary write-only batch processing phase where evidence content is extracted and analyzed. As it is usually acceptable that outcomes be available only at the end of the process and that accesses to results made by clients will be mandatorily read-only, we do not expect consistency issues of the database. Therefore, NoSQL technologies that guarantee tunable eventual consistency measured in a milliseconds scale could be employed. This will be when one can foretell benefits in terms of performances, flexibility and scalability compared to RDBMS solutions.

Modularity: AlmaNebula will be a hosting environment for pluggable modules that, upon registration, will be executed and terminated according to a user defined pipeline. Not only will the framework be able to launch modules components, but also to offer some baseline facilities like security, inter-module communication and logging. A module is to be meant as an opaque container of functions which have a predefined common structure, perform related activities and are exposed in a controlled way by means of an interface layer. Module development should not be based on proprietary schemes, but rather on well known solutions, so to attract a wide audience of programmers which could easily reuse their knowledge. In this respect, the OSGi architecture [13], a set of specifications that define a dynamic component system for Java, is a notable example, even if the benefits of a modular approach are programming language independent.

Openness: One of the most important restrain factor to a widespread adoption of cloud services is the fear to be locked into proprietary technologies. Instead, a framework which is based from ground up on open standards and possibly on open source would increase the overall level of trust of all parties involved. An open architecture is more easily portable, interoperable, inspectable and subject to contributions. This would bring an important added value, especially to a digital forensics platform that should enable all stakeholders to reproduce all operations in the most forensically sound way.

4.2. Requirements

Cloud service model: from the final user's perspective, a Software as a Service model is to be selected. Customers interface will be a web application accessible by any browser or custom apps running on desktop/notebook computers or mobile internet devices. Conforming to the Cloud's philosophy of service programmability, platform features will also be directly exposed, for example by means of SOAP based or REST web services. The backend architecture can be either physical or virtualized, even if the latter solution adds an higher degree of consolidation and flexibility in the view of a possible future migration or integration with a third party IaaS. By means of a private/public network (N), forensics users (FU) access a cloud application (CA) running in virtual machines (VM) managed by a service provider (SP). These VMs are in turn hosted in a infrastructure, placed on or off SP's premises, under the control of a cloud provider (CP). SP and CP could be different entities or belong to the same organization: no assumptions will be made in this respect, as long as a private/community deployment is enforced, in order to relax the protection mechanism that would be needed by considering a fully public counterpart. It must also be added that some arguments may exist against the adoption of a Platform as a service model for this kind of applications, as PaaS engines usually enforce a strict security model that confine user applications in a sandbox, with limited access to OS features and restricted possibilities as to sub process spawning.
or response times, among others. Furthermore, the risk of locking into proprietary technologies is still remarkable. Figure 1 shows the conceptual service model:

![Fig.1 Conceptual service model](image)

- **Alternative analysis**: There are times where accuracy could be deliberately traded for speed, for instance when it is imperative to achieve a very swift overview of a digital evidence content or, maybe, to analyze the same set of evidences with a different software just to timely increase the level of information recall. Following the directory structure, as file system aware processing libraries usually do, translates on many time consuming movements of magnetic HD heads during seek operations. Conversely, processing strategies like *stream based disk forensics* [14] efficiently read the evidence material from start to end as a byte stream and extract files by performing a recognition based on known tags or regular expressions. This could be a less complete, but quicker option to analyze a forensic image, especially in presence of unknown or damaged file systems. AlmaNebula design therefore requires that practical implementations give the user the opportunity to select when favoring completeness or speed of content and metadata extraction from digital evidences. Even better, this could be considered on a per evidence basis in order to account for evidence storage systems made of modern solid state disks where heads seek penalties do not apply [9].

- **Information extraction**: On top of traditional information retrieval (IR) techniques, that entail a deep domain knowledge as the investigator is required to know in advance what to look for to feed the search engines, it’s worth considering an information extraction (IE) layer, where the same data can be viewed from a different unexpected perspective. This is where, without user interaction, named entities like family names, emails or organizations are extracted and linked by means of natural language processing (NLP) algorithms trained on specific corpora or where documents are clustered together according to natural similarities detected using statistical properties of the text [15]. Applying unsupervised IE techniques may offer further guidance to investigators because potentially interesting documents somehow autonomously ‘pop up’ to his attention.

- **Simplified interfaces**: The need to avoid overwhelming the investigator attention calls for captivating and intuitive interfaces that waive to technical details of data in favor of knowledge management such as automated link analysis, cross correlation and zooming-in to reduce information overhead [10]. No raw content display, logical partition information or directory browsing with screens bloated with files that do not bring any immediate knowledge contribution will be present in AlmaNebula’s dashboard. Instead, an alternative approach may consist in presenting the user with baskets belonging to predefined general categories (documents, email messages, chat conversations, multimedia and so on). Additional containers will reorganize the information according to NLP algorithms or statistical properties of documents. Inside every basket each item could be still displayed in a tabular manner, but with a few properties in addition to the bare name. Nevertheless, it must be always possible to visualize full details, for example by mouse hovering.

- **Case management**: AlmaNebula will present enhanced case management features compared to the missing or really basic possibilities offered by the most part of the mentioned tools:
  - some evidence details such as acquisition hashes should be populated automatically by parsing log files, if available, in the most widespread formats (e.g. Access Data FTK Imager). The case
itself and every evidence that belongs to it should bring along also its history in terms of multimedia or documental content (e.g. pictures or written reports);
- as far as possible, in addition to the most common disk image file formats, the platform should be fed with the most complete variety of data packages. For example, in presence of network captures, import modules should be able to parse high level protocols, extract relevant stream content (such as web pages or email messages) and metadata (e.g. date/time or ip addresses);
- a role based case handling policy is to be enforced and system administrators will create users accounts or import them from an existing directory service. The Case Owner role will have full rights, whereas the Investigator role will decide what to extract from evidences and will have a read only access privilege shared with Stakeholders. Roles scope will be a single or multiple cases and further rights could be granted or revoked by case owners.

- **Security:** The threat model considers a private/community deployment and assumes that no harm can come from insiders, notably system administrators that could observe the state of a VM from the outside (e.g. memory space) by means of virtual machine introspection (VMI) tools [16]. Given the trust relationship with the computing environment, guest VMs are assumed globally integer at setup and exposed to possible risks of cyber attacks only when a connection to a corporate network is operated. A set of minimum security requirements is specified as follows:
  - **encryption:** data must be protected with strong encryption schemes, preferably based on standard algorithms like AES, when in transit and optionally at rest;
  - **user multifactor authentication:** FUs must log in by means of a multifactor authentication schema, that is based not only on what user knows, but also on what user has or is. For example, a simple two factor implementation could enforce a traditional username/password couple backed up by a an hardware token compliant to HMAC based One-Time Password (HOTP, RFC 4226);
  - **evidence content tampering control:** files content and properties must be hashed and signed upon extraction from disk images so that, if performance penalty is tolerable, every data handling operation can be preceded by a verification of genuineness to avert the possibility that data were tempered with;
  - **audit trail:** every effect stemming from users interaction with the platform, from login to evidence handling or processing must be documented and recorded in a detailed audit log which should be signed, timestamped (if possible) and cannot be directly altered through the user interface. An operation log where all steps performed by the platform following user instructions must also be produced. Logs may have more than one verbosity level, must be rotated and kept safe according to corporate security policy.

5. **Architecture & Wrap-up**

5.1. **Architecture**

*AlmaNebula’s* main building blocks can be described as follows:

- **Cloud OS:** The Cloud OS plays the fundamental role of fabric controller as it interacts with Virtual Machine Manager (VMM) to direct VMs behavior. It may offer ancillary services such as device block store that can be attached to VM instances (much like USB external drives) and object storing, used for instance for backup purposes. An API exposes Cloud OS features to a Management Dashboards in order to let administrators monitor and orchestrate the operations of each and every component, such as running and metering VMs. It is desirable that the Cloud OS be VMM agnostic.

- **Virtual Machine Manager:** A type I VMM (bare metal) setup [17] that runs directly on commodity hardware has been selected, but type II VMMs (hosted) that lie on top of an host OS are also possible. For portability purposes, VMM should support well documented or, better, standard VM file packaging like Open Virtualization Format.
VM: VMs host the distributed storage which is created on top of their virtual hard disks. They should be guest OS agnostic. During processing, VMs are started according to the chosen Map function.

Storage layer: A database is a convenient way to integrate modules pipelining and store metadata of evidence files in order to leverage filtering and sorting capabilities. File content can be inserted in the database too, even if it could prove more handy using the storage space of the file system, where some existing tools like search engines can process them directly without prior extraction. A distributed file system can be created as an abstraction layer on top of the database, inserting a mediation module that converts POSIX calls such as open() or read() into SQL queries. In the storage abstraction layer the functionalities to shield applications from the internals of DBMS should also find their rightful place. If a regular distributed file system is to be preferred, the choice must privilege highly available solutions that could be installed on top of modern journaled file systems like Ext4. Hadoop's file system HDFS could be an option much like Ceph or Gluster FS.

Cloud application: the cloud application will be made of pluggable modules that will be pipelined under the supervision of the control logic to reflect user configuration. The Content & Metadata ingestion module will preliminarily populate the storage and prepare the ground to the following modules, notably: 1) Information Retrieval (IR) that will perform text indexing and pattern searching according to exact matching and regular expressions; 2) Information Extraction (IE) which will extract named entities and will cluster documents; 3) Super-timeline reconstruction that, overtaking the limit of the traditional timeline reconstruction based just on file last modification date/time, will also dig into several log files to rebuild a more exhaustive picture of the actions that took place in a device. More modules can be added to perform specific functions.

Cloud application API: a web services based API will expose platform capabilities to web applications and mobile apps after a strong user authentication has been performed by the Security module as described earlier in the requirements section. An overview of the platform proposed as IaaS is sketched in Figure 2:

Fig.2 AlmaNebula's simplified architecture

5.2. Wrap-up

The most important design goals and requirements that fuel the platform have been introduced and are all resumed in Figure 3. Discussion also showed that, compared to some well established forensic solutions discussed earlier, some feature improvements can be proposed that would be valuable for non-
technical final users. In the testing phase it will be interesting to monitor how evidence processing time varies according to the number of collaborating VMs. Adapting the technical and legal requirements of computations that are to be outsourced to commercial CSP will be a major development direction.

6. Conclusion

This paper aimed at providing a contribution to architecting a novel forensic application, which exploits the computational power and storage capacity of collaborating commodity machines to process huge collections of digital evidences. Software development of a working prototype is underway.

Figure 3. Summary of AlmaNebula’s design goals and requirements

Acknowledgments

The author wishes to thank the Italian Arma dei Carabinieri for its support in producing this paper.

References