Towards a Big Data System Disaster Recovery in a Private Cloud

Victor Chang,
School of Computing, Creative Technologies and Engineering,
Leeds Beckett University, Leeds, UK
V.I.Chang@leedsbeckett.ac.uk

Abstract— Disaster Recovery (DR) plays a vital role in restoring the organization's data in the case of emergency and hazardous accidents. While many papers in security focus on privacy and security technologies, few address the DR process, particularly for a Big Data system. However, all these studies that have investigated DR methods belong to the "single-basket" approach, which means there is only one destination from which to secure the restored data, and mostly use only one type of technology implementation. We propose a "multi-purpose" approach, which allows data to be restored to multiple sites with multiple methods to ensure the organization recovers a very high percentage of data close to 100%, with all sites in London, Southampton and Leeds data recovered. The traditional TCP/IP baseline, snapshot and replication are used with their system design and development explained. We compare performance between different approaches and multi-purpose approach stands out in the event of emergency. Data at all sites in London, Southampton and Leeds can be restored and updated simultaneously. Results show that optimize command can recover 1TB of data within 650 seconds and command for three sites can recover 1 TB of data within 1360 seconds. All data backup and recovery has failure rate of 1.6% and below. All the data centers should adopt multi-purpose approaches to ensure all the data in the Big Data system can be recovered and retrieved without experiencing a prolong downtime and complex recovery processes. We make recommendations for adopting "multi-purpose" approach for data centers, and demonstrate that 100% of data is fully recovered with low execution time at all sites during a hazardous event as described in the paper.

Keywords— Disaster Recovery (DR); TCP/IP baseline, snapshot, replication, multi-purpose DR approach, performance measurement

1 Introduction

BIG Data technologies, services and its adoption have been a topic of discussion in the past few years. According to International Data Corporation, the overall created and copied data in the world was 1.8 zetabytes in 2011 and the volume would be estimated to have nine times more than current values within five years [1]. Apart from volume, other characteristics of Big Data such as velocity, variety, veracity and value should be investigated and balanced. Velocity refers to the speed of the data processing the big datasets or large volume of data. Variety refers to the different types and formats of data exited in the Big Data services. Veracity refers to ways to keep data clean, having good quality and accurate after removing abnormality and biasness. Value refers to the use of Big Data services that can produce positive business values such as increased business opportunities and improvement in efficiency to the organizations that adopt Big Data. Several trials have confirmed that Big Data can make positive impacts to organizations that adopt it and can produce a significant impact when Big Data integrates with Cloud Computing [2-4]. Chen et al [2] provide a detailed survey about Big Data, which includes the literature review, related technologies, characteristics of Big Data, technologies and future trends of Big Data. Chen and Zhang [3] describe the data-intensive applications, challenges, techniques and technologies of Big Data and they summarize seven principles of using and managing Big Data through their review and technical synthesis of Big Data technologies. Agrawal et al. [4] present Big Data and Cloud Computing current state and future opportunities. They describe a few Cloud Computing technologies that be used by Big Data services. Chen et al [5] have a different perspective. They explain the journey of business intelligence development over the previous thirty years and present the case that the development has been focused on Big Data and then shifted to Big Impact.

There are other challenges for Big Data adoption, particularly security, which is an important aspect for all organizations involved. Armbrust et al [6] define technical challenges and the security issues to be resolved for Cloud Computing and also Big Data. One aspect is ownership and privacy of the data stored in the Cloud. Data in the Cloud should have a clear ownership definition, and not be shared or accessed by users without authorization. Legal and data compliance fulfilling governments and regional/international laws need to be met. The ownership of data is not restricted by whether the users or service providers own the data in the Cloud and who is accountable for data lost [7, 8]. Data management in Public or Private Clouds should include a backup or contingency plan, so the impacts can be minimized if disaster strikes. Khajeh-Hosseini et al [9] show in their surveys and case studies of Cloud Computing adoption which consider the costs, benefits and risks involved, that outsourcing can move operational and adoption risks to

the service providers, but in reality this does not always eliminate the risk of data loss. Their recommendation is also applicable to Big Data system adoption. It has been suggested that using a Private Cloud approach can lead to better management of data and resources [10] but it cannot prevent data loss due to Data Center hazards such as fire, flood, earthquakes, blackout, hacking and sudden death of hard disks (due to heat, lifespan or power failure).

The risk of data loss applies regardless of whether organizations put their data into public or Private Clouds, or both. Data can be lost, or corrupted at any stage of Cloud Computing adoption. Hence, maintaining data accessibility, control and management requires a long-term strategy and a complete solution, including a plan for backing up data regularly and disaster recovery to ensure data are restored promptly to enable business continuity.

1.1 DISASTER RECOVERY LITERATURE AND OVERVIEW

While a significant number of papers focus on security and privacy with regard to Cloud Computing, the number of papers focusing on disaster recovery (DR) is relatively small. DR should be adopted by all organizations [11], and not just for Cloud Computing. Even so, the speed of putting data and updating data in the Cloud is much faster than desktop systems [12], and the use of Cloud also means that users can be anywhere and at anytime. In other words, organizations should focus on improving the efficiency and quality of data recovery. One indicator is the percentage of data lost and damage in the DR process should be lower than using desktops [13, 14].

Disaster recovery is not a problem for cloud service providers but every organization that uses cloud [15]. If data are irretrievably lost, this may have negative impacts on the organization affected such as financial loss and loss of time to reproduce or regain data. We argue that this is an ongoing challenge for adopting Cloud Computing, whether it is public, private or hybrid cloud. Of course, it is more relevant for those adopting Private Cloud, since there is no outside service provider to share the responsibility.

The existing literature on DR process is presented as follows. Pokharel et al [16] explain the drawback of existing DR practices, and suggest that factors such as high cost, loss of energy, undesirable downtime, under utilization of infrastructure and low reliability and availability prevent successful DR process delivery. While these factors are important, their approach is based on "single-basket", which means one technique and one site to handle disaster recovery [16, 17]. Consider the situation should the emergency backup server fail to respond in the event of fire, will all the data be lost, including the rescue data?

Pokharel et al [16] propose a geographical redundancy approach (GRA) to minimize these risks, and explain their architecture, including each component used. However, GRA is only a theoretical model, and it is not obvious that it can be implemented successfully for organizations adopting Private Cloud. Additionally, there is insufficient information about the background, set of experiments, how to obtain these results and their detailed descriptions of result analysis. Wood et al [16] provide a comprehensive review of current DR literature and practices, including each factor influencing DR process. They also explain that there are three types of DR mechanisms, and the importance of failover and failback. Their contribution is focused on the economic benefits of DR processes. They first explain the cost per day model and make comparisons based on their Public Cloud. The cost of replication is $2.04, $0.54 and $1.22 for server, network and storage respectively a day. These figures are reliable for server and network but the cost for storage warrants further scrutiny. For example, the current daily cost of Amazon EC2 Storage pricing on their website [17] is from $0.011/GB to $0.095/GB depending on performance. While authors published their work three years ago, the lowest pricing was very likely the one (they did not say that in their paper) since they only spent $1.22 per day. The cost became $36.6 per month on average, excluding failover costs, which were $52.03 a day, or $1560.90 a month. We would argue that costs for failover, or data recovery during at-risk periods must be taken into consideration. Based on Woods et al [17] suggestion, a large organization can spend about $1,600 a month just for 30 GB of DR process that includes data recovery during at-risk periods.

Subashini and Kavitha [18] explain the significance of Cloud security including disaster recovery. They acknowledge the importance of DR process and only provide the overview and the related literature. There are no any empirical studies and thus their proposal does not have a strong support and recommendation. Snedaker [19] focuses on the organization's strategies, steps and policies to perform DR and suggest ways for organizations to restore their data through a consolidated process built in place. However, Snedaker does not focus on setting up Cloud platforms for DR. Since more organizations have put their data on the Clouds, there is a growing concern for the Cloud-adopting organizations to have DR processes for their Cloud services [15].

Amongst the limited literature that provides empirical studies in DR processes in the Cloud, Wood et al [20] present a more comprehensive synthesis than the previous recommendations [16-19]. Wood et al [20] then propose their PipeCloud, based on the recommendations from an earlier paper [17]. They state the DR challenges involved, and recommend replication strategies and ways to minimize latency. They explain definitions of their Recovery Point Objective (RPO) and Recovery Time Objective (RTO), the former refers to the acceptable amount of application data that can be

lost; and the latter is the amount of downtime that is allowed before the system recovers. They further propose their PipeCloud based on RPO and RTO, and explain how to synchronize both, by introducing "pipeline synchronous replication". They demonstrate how their PipeCloud can work under the conditions they describe, and explain the implementation and results of experiments. They divide their data with read and write processes, and show their PipeCloud can handle the DR process. However, there are two main limitations based on our analysis. First, they assume that there is only one place to move rescued data, and that is why they introduce synchronization to streamline both the source (data to be rescued) and destination (data rescued, restored and moved). We argue that DR process should be distributed over multiple sites, or techniques for full data recovery. The PipeCloud solution is like putting all your eggs into one basket, and if that basket is broken, everything is lost. Second, more details of their set of experiments, any algorithm or syntax of code should be provided.

Sengupta and Annerrvaz [21] present their multi-site DR data distribution, including their system architecture, theories, data center details and costs involved in the DR process. They also have the consolidated theories for PRO and RTO with experimental results showing a low costs for the DR process. However, the size of data for backup and storage is very low, since each data center only takes between 2 and 4 terabytes. It looks more like each data center is a server room with storage servers that offer terabytes of services. In the modern data centers, the volume contains a few petabytes with more complex data distributions than Sengupta and Annerrvaz's proposal, which does not show how to overcome some of the challenges and requirements for volume, velocity, variety, veracity and value.

Our novel contribution of this paper is to demonstrate a multi-purpose approach for disaster recovery, since the existing literature suggests only single approach has been adopted. Our approach can meet the requirements for Big Data systems and services that can fulfill requirements for volume, velocity, variety, veracity and value, with all data restored and updated in four sites. This is particularly useful for service providers who manage cloud data centers.

1.2 An overview to our proposed approach and structure of this paper

Our proposed approach takes on multi-site and multi-technique approaches, ensuring that if one method does not succeed, there are more methods that can retrieve and restore data on time. Our proposed solution takes a comprehensive approach that can deal with daily backup of terabytes of data [9]. Additionally, our DR process is based on data retrieval with multi-techniques and multi-sites to ensure that DR process can be smooth and successful each time. We also focus on investigating the performance and the failure rate of losing data. We offer the traditional TCP/IP method, snapshot recovery (for Virtual Machines, VMs) and a hybrid replication solution based on TCP/IP method and snapshot. Mirror sites and tapes are also used to ensure a very high percentage (e.g., 99.9%) close to full recovery can be achieved. In this way, we can protect our data against all forms of hazards, and safeguard valuable data owned by the adopting organizations for years in a Private Cloud environment. We focus on a Big Data system in a Private Cloud so that the organizations can have a better control and a full ownership of the data for the backup and DR.

For this paper, we demonstrate our proposed approach as follows. Section 2 describes our motivation, the technology used, system development, open Virtual Machine Format (OVF) and architecture to deploy DR. Section 3 presents system development and process required for DR, and starts with core code syntax with their explanations for their significance and functionality. It includes the syntax for traditional TCP/IP method, anti-intrusion protection, snapshot recovery, replication and data recovery with decryption. Section 4 demonstrates all the essential experiments investigating performance and failure rate associated with the DR process. The experiments are focused on the traditional TCP/IP method, including data migration of between 1,000 and 10,000 files, and large single files between 100 GB and 1 TB, and the failure rate of data migration in these two sets of scenarios. Experiments involved with snapshot recovery and replication are also discussed. Section 5 describes detailed experiments between single and multi-purpose approaches and results support the effectiveness of multi-purpose approaches for disaster recovery. Section 6 presents the discussions and conclusion of this paper.

2 The deployment and architecture for the disaster recovery

This section presents the background work for disaster recovery (DR) in a Big Data system in the Private Cloud.

2.1 Motivation

Both Guy's and St Thomas' NHS Trust (GSTT) and King's College London (KCL) have a data center at the University of London Computing Centre (ULCC). Each holds as many as 2,000 physical high-end servers. They also provide fiber optics and high-speed switch network infrastructure to allow advanced experiments to use network speeds of up to 10 gigabits per second (GBps). It is the interest of both the management and scientists to use a facility located at the GSTT and ULCC. This allows backup process approval to be completed quicker, and is a good exercise ahead of the merger of both groups under a new umbrella organization, King's Health Partners. The facilities at ULCC have better IT and staff support. Additionally, there was a funding opportunity in the Department of Health, UK in 2008 to design and build a system for proof-of-concept and improvement of efficiency, including a Big Data system solution. The pro-

posed solution was a testbed for NHS trusts in London.

2.2 THE CLOUD SOLUTION OVERVIEW

Supported by an organization called King's Health Partners in 2008, GSTT and KCL were able to work together on projects to implement a Big Data system and deliver a Big Data Storage Area Network (BDSAN) as a service that can support at least 10 petabytes of services on daily basis. Both institutes have used the Cloud strategy, design, implementation and user support proposed by this paper.

Table 1: Selections of Technology Solutions

Technology selections	What it is used for	Vendors involved	Focus or rationale	Benefits or impacts
Network Attached Storage (NAS)	To store data and perform automated and manual or personal backup.	Iomega EMC Lacie Western Digital HP	They have a different focus and set up. HP is more robust but more time-consuming to configure. The rest is distributed between RAID 0, 1 and 5	Each specific function is assigned with each NAS. There are 5 NAS at the GSTT/KCL site and 3 at the Data Centre, including 2 for archiving.
Infrastructure (networking and hosting solution)	Collaborator and in-house	Data Centre at University of London Computing Centre (ULCC)	Some services need a more secure and reliable placement. University of London Data Centre offers 24/7 services with around 500 servers in place, and is an ideal for hosting solution	Amount of work is reduced for maintenance of the entire infrastructure. It stores crucial data and is used for archiving (backing up historical data and backing up the most important data automatically and periodically)
Backup applications	Third party and in-house	Open Source Oracle HP Vmware Symantec In-house development	There is a mixture of third party solutions and in-house development. HP software is used for high availability and reliability. The rest is to support backup in between NAS systems. Vmware is used for virtual storage and backup	Some applications are good in a particular service, and it is important to identify the most suitable application for particular services
Virtualization	Third party	VMware VSphere and Citrix	It consolidates IaaS and PaaS in private cloud deployment	Resources can be virtualized and saves effort such as replication
Security	Third party and in-house Fined-grained model security	KCL/GSTT Macafee Symantec F5	Security is based on the in-house solution and vendor solutions and is focused on secure firewall and anti-virus	Remote access is given to a list of approved users

From the perspective of healthcare executives, for a Big Data system service to be a success and demonstrate better performance than a non-Big Data system service, it must deliver improved efficiency whilst managing to keep the rate of job and backup failures low. They require a strategic plan to recover data quickly and efficiently when unexpected events such as fire, hard-disk corruption and malfunction of air-conditioning systems (that causes over-heating of the servers) happen. The proposed Cloud solution involves the design and deployment of a Big Data system in the Private Cloudacross three sites located at Guy's Hospital, King's College London (KCL) Medicine and the University of London Computer Centre (ULCC), the latter of which has a data center holding up to 600 high-end servers for all the universities of London (and rest is for BT), providing Big Data system, clusters and virtualization services for the entire

University of London.

The Big Data system is a Private-Cloud SAN architecture made up of different Network Attached Storage (NAS) services, where each NAS is dedicated to one specific function. The work involves integrating software and cloud technologies from commercial vendors including Oracle, VMWare, EMC, Iomega and HP. This is to ensure a solid infrastructure and platform is available. Design and deployment is based on the user group's requirements and their research focus. Selections of technology solutions are essential for Big Data system development, as presented in Table 1. To ensure heterogeneous data, all data must be in the same size. Often scientists have undertaken their experiments that could generate lots of data and all have to be zip up as a in a unit of 1 GB or 10 GB to ensure they can be backed up without causing problems in dealing with size variation during the DR process. Descriptions for each NAS system at ULCC are as follows:

- One NAS is used as a central backup database to store and archive experimental data and images.
- The other two advanced NAS systems are customized to store and archive valuable data.

Additionally, there is one Bioinformatics SAN backed up at the ULCC.

2.3 THE DESIGN AND DEPLOYMENT OF THE INTEGRATED BIG DATA SYSTEM

The integrated Big Data system designed to provide functionality and services for archiving, data storage, data management automated backup, data recovery and emergency recovery, which are considered as Platform as a Service (PaaS). The Big Data Architecture uses two concurrent platforms. The first is based on Network Attached Storage (NAS), and the second is based on the Storage Area Network (SAN). The NAS platform provides great usability and accessibility for users. Each NAS device may be allocated to a research group and operate independently. Then all the NAS devices can be joined up to establish a SAN, which can consolidate an organizational backup platform and can improve capabilities and performance. SAN allows data to be kept safe and archived for a long period of time. The design of SAN focuses on SCSI (Small Computer System Interface), which offers dual controllers and dual networking gigabit channels. Each SAN server is built on a RAID 10 system to offer a good performance like RAID 0, but also has mirroring capability like RAID1. The integrated Big Data system can achieve the following functions:

- Performance improvement and monitoring: This allows tracking the overall and specific performance of the Big Data cluster, and also enhances group or individual performance if necessary.
- Disk management: When SAN system pool is established, it is important to know which hard disks in the Big Data system support which servers or which user groups.
- Advanced features: These include real-time data recovery and network performance optimization.

Virtual Machines (VMs) are actively used at all sites of the NHS Big Data system. VMs play essential roles in the disaster recovery (DR) that requires highly-organized scripts and IT management policy to rescue data and reduce the organizational impacts. VMX format and the Open Virtual Machine Format (OVF) are used in VM management. OVF is a hypervisor-neutral standard for describing, packaging and distributing virtual appliances, and can be used by products manufactured by other firms, including VMware, Citrix, Cisco, Microsoft, HP and Dell. An OVF package consists of an XML document known as the OVF descriptor, which specifies metadata about the virtual appliances and virtual disk files. The benefit of doing so is that an OVF can describe a set of virtual machines or virtual systems, and this helps system architect to use scripting to manage all virtual machines.

There are other terms to define. For example, the "OperatingSystemSection" describes aspects of what the runtime environment can expect from the virtual system once the DR process is executed. The "VirtualHardwareSection" specifies aspects of what the virtual system requires from the runtime when the DR process is kicked off. To explain how OVF can work, a complete example is provided as follows.

2.4 EXAMPLES TO DEMONSTRATE OPEN VIRTUAL MACHINE FORMAT (OVF) IN THE DISASTER RECOVERY

A proposed XML section type, DisasterSection, is described here as an example to support DR. The focus here is replication, which means retrieving and obtaining data from the backup Big Data system servers presented in Section 2.2 and 2.3. All the files are backed up and retrieved from secure ports such as 22 for secure FTP and 443 for secure HTTPS. Instead of displaying IP addresses in the traditional method, the IP addresses in all virtual machines are assigned at runtime, and there is an OVF ID that handles processing of the DR request.

For example, the syntax can be ovf:id ="disasterrecovery" presented in Table 2. All the OVF IDs can be mapped to the required IP addresses when a VM is deployed. This allows DisasterSection to describe not just a single VM behavior, but expected communications and actions between VMs required for DR. Another feature in Table 2 shows ovf:required="true". This means that DisasterSection is required to prompt the action. What triggers DisasterSection is when the confirmation for fire hazards or data center failure from the authorized staff at the ULCC is received (which was a fast process within minutes if that happened). This ensures that actions for DisasterSection can only go ahead when it is confirmed by the authorized staff at the ULCC but not the system architect of this NHS Private Cloud.

Table 2: The DisasterSection for emergency servers

```
<ns:DisasterSection ovf:required=”true” xsi:type="ovf:DisasterRecovery_Type">

<Info> Disaster Recovery for NHS private cloud </Info>
<Rule>
   <Info> Retrieve data for disaster recovery </Info>
  <Protocol> tcp </Protocol>
  <DstAddr ovf:id=”disasterrecovery” />
  <DstPort> 22 </DstPort>
  <DstPort> 443 </DstPort>
  <SrcAddr> any </SrcAddr>
  <SrcPort> any </SrcPort>
 </Rule>
 <Rule>
  <Info> Connection to emergency backup server </Info>
  <Protocol> tcp </Protocol>
  <DstAddr ovf:id=”disasterrecovery” />
  <DstPort> 3306 </DstPort>
  <SrcAddr ovf:id=”disasterrecovery” />
  <SrcPort> any </SrcPort>
 </Rule>
 <Origin>
  <Info> Firewall protection for all VMs </Info>
  <DateAdded> 2014-01-18 </DateAdded>
  <AddedBy name=”Administrator” role=”creator”</>
 </Origin>
```

Other features are presented as follows. The tag, Rule, is to specify and execute the policies set by the author. The tag, Info, is to explain the actions to be taken under “Rule”. The action is to retrieve data from the backup server (which backs up data on daily basis), and then recover all backup in two emergency servers, where one is located at the ULCC and one is located at another server room at the KCL Medicine. Two emergency cloud servers means that if accidents such as fire happen at one Hospital (and affects the networks, either to KCL Medicine, or ULCC), it has at least another server to recover the retrieved data from the ULCC. In addition, the content of the original backup servers can be unchanged, and does not get involved in recovering the data, which can consume a large volume of disk space. Tape is also used and backed up weekly, and is useful in case that network failure happens presented in Table 3. Additionally, tape backup system supports TCP protocol other than backup by physical tapes. Port number 3494 is the port number for the tape systems.

Table 3: The DisasterSection for the tape system

```
<ns:DisasterSection ovf:required=”true” xsi:type="ovf:DisasterRecovery_Type">

<Info> Disaster Recovery for NHS private cloud </Info>
<Rule>
   <Info> Retrieve data for disaster recovery </Info>
  <Protocol> tcp </Protocol>
  <DstAddr ovf:id=”disasterrecovery” />
  <DstPort> 3494 </DstPort>
  <SrcAddr> any </SrcAddr>
  <SrcPort> any </SrcPort>
 </Rule>
</DisasterSection >
```

There is an additional XML code to enforce security shown in Table 4. The rule 43 is enabled since it supports the plan presented in Table 2 and Table 3, and also ensures that all the existing systems are protected from intrusion by Cisco security, just in case that the serious accidents are caused by unauthorized hacking rather than natural disasters or fire.

Table 4: The DisasterSection for the tape system

```
<ns:DisasterSection ovf:required="true" xsi:type="cisco:IntrusionProtection_Type">
<Info> Intrusion Protection for NHS private cloud </Info>
<Rule>
  <Info> Retrieve data for disaster recovery </Info>
  <Protocol> tcp </Protocol>
  <DstAddr ovf:id="disasterrecovery" />
  <DstPort> 3494 </DstPort>
  <SrcAddr> any </SrcAddr>
  <SrcPort> any </SrcPort>
 </Rule>
</DisasterSection >
```

2.5 THE ARCHITECTURE TO DEPLOY DISASTER RECOVERY

Figure 1 shows the architecture to deploy disaster recovery (DR), which has been explained between Section 2.1 and 2.4. When the "DiasterSection" is on, upon the confirmation from the authorized staff, the Big Data backup servers begin the DR process. First, the data is sent across to two emergency backup servers located at different sites. Second, the data is also backed up from the Big Data backup servers to local backup and tape systems located at another building. Third, the data is received at the two emergency backup servers. Forth, the data is received at the emergency backup servers and local servers begin with the recovery process. Monitoring at the Big Data backup servers is used to protect existing backup servers from hacking if the cause of DR is not natural accidents or fire.

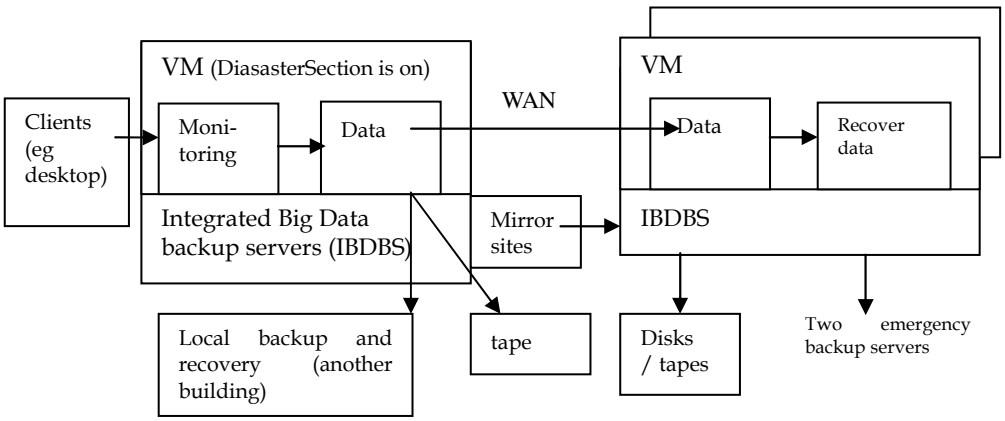

Figure 1: The architecture between KCL and ULCC to deploy disaster recovery

Some data received at the emergency server are encrypted, and that is why recovery process is required for the following:

- To check the data is not corrupted and in a good healthy status.
- To decrypt the data
- To store the data safely

Mirror sites have the full version of backup taken at 7 pm of each day, and can be triggered to help with DR process. Additional details will be presented in Section 3.

2.6 MULTI-SITE DISASTER RECOVERY STRATEGIES

This section describes the multi-site DR strategies. The main DR venue is the University of London Compute Center (ULCC) which backs up data from the KCL Medicine to ULCC. However, if ULCC is unavailable due to the accidents such as fire or flood despite of its low possibility, it has the risk to lose data and not being able to recover data on time. The best strategy is to offer multi-site DR that backs up all the data simultaneously and does not take a long time. The sites include Southampton and Leeds to ensure that all files can be backed up in parallel with ULCC. Due to the distance involved, full backup by TCP/IP is not a preferred method for concern such as data loss and drop in quality of service if the process involved takes hours each time. The recommended approach is to use snapshot, which captures important record of files and records at least twice or three times a day. Snapshot is a reliable method and can capture a high majority of data for backup and can be completed much quickly than the full backup by TCP/IP method. Experiments involved with snapshots with multi sites will be presented in Section 4. Figure 2 is a diagram to show the

architecture between ULCC and data and resource centers at Southampton and Leeds. There are two data centers in Southampton, whereby one is based at the University of Southampton and one is located at the author's home. The resource center in Leeds has received snapshots twice a day and snapshots can be successfully restored and replicated. The multi-site approach can ensure that no data has been missed for backup and restoration.

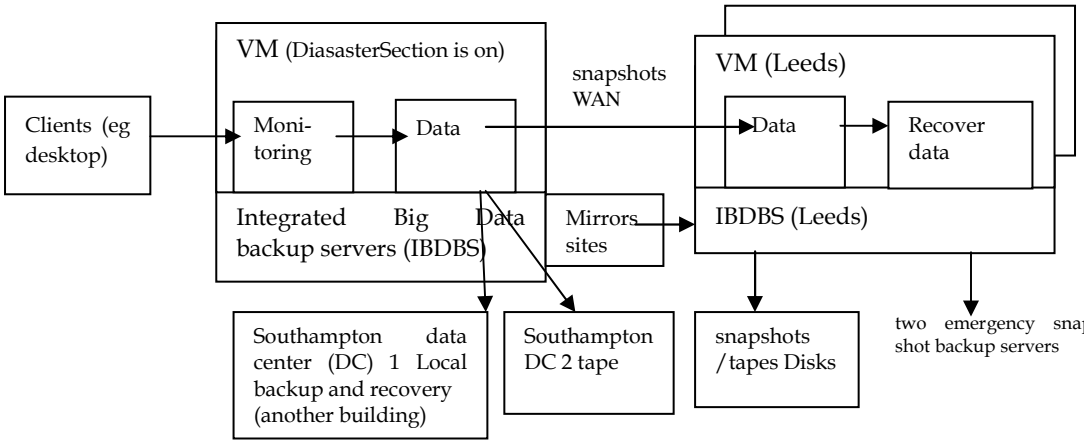

Figure 2: The architecture between ULCC and Southampton and Leeds data and resource centers

2.7 THE DEVELOPMENT OF APIS TO ACHIEVE THE MULTI-PURPOSE DATA RECOVERY

The challenge is to make all four sites in London (KCL and ULCC), Southampton and Leeds to fully restore data simultaneously and ensure that all four processes are ongoing without interfering with one another but consolidating with each other. In order to demonstrate the multi-purpose DR, Application Program Interfaces (APIs) are developed. The advantage of using Cloud Computing oriented APIs is to allow the architect and users to use anywhere with the internet access and with a short line of codes, in most cases, just one phrase like in one of our publications to demonstrate business intelligence as a service in the Cloud [22]. The development for multi-purpose approach also adopts the same strategy to use short but effective API to execute backup and data recovery. This section presents the system design for "restore" API. There are also location-based commands.

- restore(all): To restore all the data in all sites and take the DR process by the default methods including TCP/IP, snapshot and replication.
- restore(optimize): The most crucial backup is between KCL and ULCC and thus accelerating the speed of this site takes the priority.
- restore(London): Restore data to London first and wait for the next actions.
- restore(Southampton): Restore data to Southampton first and wait for the next actions.
- restore(Leeds): Restore data to Leeds first and wait for the next actions.
- restore(TCPIP): Restore data by TCP/IP method to all sites.
- restore(snapshot): Restore data by snapshot method to all sites.
- restore(replication): Restore data by replication method to all sites.
- restore(check): Check the status of the DR process
- restore(stop): Stop the DR process immediately
- restore(restart): Restart the DR process
- restore(report): Get the report of the DR process whether it is successful or failed.

3 System deployment and process required by disaster recovery

This section describes different types of system development and process required by disaster recovery (DR). The content includes the code syntax to proceed with the DR, the security process involved and related system process involved. It then presents the multi-purpose approach. It is a state-of-art technology to ensure all sites in London, Southampton and Leeds have data backup and recovery in synchronization. Understanding each technique is important before presenting the multi-purpose approach.

3.1 THE CORE CODE TO DEPLOY DISASTER RECOVERY IN TRADITIONAL TCP/IP METHOD

This section explains core code to proceed with DR with the TCP/IP method, where "status(job)" is to check the status of the DR process and if 'disaster' is equal to 1, which means the DR process is kicked off as shown in Table 5. The DR process is terminated when all data from the Big Data system servers are transferred to and recovered at the

emergency servers. Explanations for other parts of the DR process are as follows.

- "record(status(job))" is to record job status.
- "rerun(status(job))" is to run DR process again in case some files are not found or not transferred.
- "report(status(job))" is to report to the system at once after rerunning failed jobs is successful.

Table 5: System administration code syntax to kick off disaster recovery

```
If (disaster == 1)
  continue (status(job));
else
  stop(status(job));
  report(status(job));  // report that there is an error
  exit
end

check(status(job))   // to check whether the disaster recovery process is
achieved
if (disaster == 0)
   complete(status(job));
   report(status(job)); // report to the SAN that everything is completed
  end
else
   record(status(job))  // record the status of failed jobs
   rerun(status(job))   //rerun failed jobs before reporting to the system
   report(status(job)); // system report is  sent off for review
end
```

3.2 THE SYNTAX TO MANAGE INTRUSION PROTECTION

This section describes the intrusion protection used during DR to ensure that the DR process is safeguarded all the times. Cisco networking administration is adopted, which uses crypto key in the Intrusion Prevention System (IPS). The core syntax includes:

```
crypto key pubkey-chain rsa
named-key realm-cisco.pub signature
key-string
```

While typing these three lines, an encrypted key-string is generated automatically to protect the data from potential malicious hack. The key-string may look like this:

```
B199BACB D3D0F94E 058FADC2 425C197E F21AF10A FBC0E516 7E0764BF 4E62053E
```

Once the key generation is done, the IPS configuration can be saved. Similar to "DisasterSection" XML tag in Section 2.5, the next step is to create a rule for IPS, followed by configuring IPS signature storage location. The final step includes IPS event notification. Their respective steps can be presented in Table 6.

Table 6: The code syntax to kick off IPS

```
ip ips name <rule name> < optional ACL>
router#configure terminal
router(config)# ip ips name iosips

ip ips config location flash:<directory name>
router(config)#ip ips config location flash:ips

ip ips notify sdee
router(config)#ip ips notify sdee
```

3.3 SNAPSHOT RECOVERY FOR VMS

The Big Data backup servers take snapshots of VMs on a daily basis, and the snapshots are archived and stored. The NHS Private Cloud holds the most recent four weeks of snapshots. So when a DR process is kicked off, it offers the recovery of the most recent snapshot. In this way, data within the VMs can be restored. The architecture diagram in Figure 1 is still applied, except data is sent as VM snapshots rather than raw data in the entire Big Data backup servers. Results and experiments will be presented in Section 4.

3.4 REPLICATIONS FOR DATA AND VMS

The DR process also includes replication, which means data (backed up on daily basis) are transferred to the emer-

gency servers from their mirror sites. In this way, data can be restored to prevent any data loss due to the events of fire and accidents. The difference between Section 3.3 and 3.4 is that Section 3.3 only backs up at a particular instance of the day, and only data within the VM. Replication backs up the entire data, including system data outside the VM (but used by the Big Data), and all VMs involved. Hence, replication is a longer process than the Snapshot recovery.

Table 7: The code syntax to initiate replication process

```
RequestValue ::= SEQUENCE {
 action INTEGER {
    suspend          (0),
    resume           (1),
    replicateNow     (2) },
 scope INTEGER {
    singleAgreement    (0),
    allAgreements      (1) },
 entryDN LDAPDN
}
```

The replication process includes these actions: To suspend replication, resume replication, or force immediate replication by a supplier server. There is also a service agreement involved. If a replication agreement is suspended, agreement queues the updates to its replica server until advanced replication is resumed for the agreement. The syntax is presented in Table 7.

The explanation is as follows. Suspend means the agreement is halted until it is passed. Resume means replication process continues. ReplicateNow means a replication agreement is waiting for scheduled replication to occur. The most commonly-used command is "resume", and it proceeds with DR process until the replication process is over. When faults happen, it can return to "suspend" until further notice, or the system architect can choose to manually change to "resume" after problem is fixed. The term "singleAgreement" means the request applies to a single replication agreement, and "allAgreement" means all requests are for all replication agreements. For the DR process, "allAgreement" is chosen to ensure that the DR process can go ahead with the minimum level of interruptions. The NHS Private Cloud has its mirror sites at another building of the Guy's Hospital, and data can be sent from the Guy's Hospital to ULCC Data Center. Section 4 will present experiments and results.

3.5 DATA RECOVERY INVOLVED WITH DECRYPTION

Referring to Figure 1 and Figure 2, the last stage involved with the DR process in the emergency server includes data recovery, which checks:

- whether any data has been lost in the data migration, and transfer the data back right before the end of the DR process.
- whether any data has been corrupted, if so, transfer the data back right before the end of the DR process.
- the number of files from the source and the destination is the same, and if not, identify the missing ones, and transfers them back right before the end of the DR process.

If any of these data cannot be recovered, data can be physically recovered from the tape, after identifying a version that has working versions of the data. An important goal for the DR process is to transfer data from the backup servers to emergency servers safely and it should be completed as soon as possible. Hence, encrypting and decrypting the data is less significant, since the sensitive data at the backup servers were already encrypted before the DR process began. On the other hand, the system architect can manually operate the encryption to take place on the protected data.

3.6 FAILOVER AND FAILBACK

Failover is the process to switch to a redundant server upon the previous active backup. In this case, the DR process looks up either the emergency servers or mirror sites as the redundant servers. Failback is the process of restoring a system in a state of failover back to its original state before failure. Our proposed approach can proceed with failover easily through the use of snapshots. In this way, if the recovery process is unsuccessful, another snapshot can be started again without interrupting the content of the files. A factor of concern when considering failback is the number of failed attempts to initiate snapshot recovery. Results will be discussed in Section 4.3.

3.7 MULTI-PURPOSE DISASTER RECOVERY INVOLVED WITH THREE ADDITIONAL SITES

While literature in [16-21] suggest that only a single method was used for each DR process, the multi-purpose DR approach has been adopted in this paper. A list of commands has been presented in Section 2.6 to facilitate the DR process, so that a single command will execute job requests and backup data to London, Southampton and Leeds in a fastest possible and the safest possible way. This includes the traditional TCP/IP method, snapshot and replication by using the restore() API.

There are two use cases. The first case is to restore everything by default. The second case is to restore to the most important data center in London by a specific method. All the code syntax is short but precise and effective. They ensure

that all data can be backed up immediately to all or specific locations, or by all the methods or a particular method that can optimize the performance. See Table 8a. Additional experimental results will be presented in Section 4.7.

Table 8a: Two cases of using restore() functions to perform disaster recovery

//case 1
If (disaster == 1)
  restore(all);
  restore(check);
  restore(report);
end;
	//case 2
	If (disaster == 1)
	  restore(optimize);
	  restore(check);
	  restore(report);
	end;

4 The experiments and results

As discussed in Section 1.1, the proposal from Wood [17, 20] puts the DR process into one basket. This may create a problem of losing the recovered data, if the rescued server is at the risk such as in fire accident. To offset the risk of losing both backup data and rescued data, we propose a "multi-purpose" approach by introducing three types of technologies. This approach has not been in literature and makes a worthwhile contribution of adopting different methods.

1. The first technology is the traditional TCP/IP method, which builds up the baseline for the DR process. It ensures that the DR process can run smoothly, with a low failure rate of losing or damaging data.
2. The second technology is based on snapshot of VMs, which offers better performance and a low failure rate (but higher than the baseline).
3. The third technology is a hybrid approach, replication for DR processes. It ensures that it inherits characteristics of snapshot by having a better performance. Replication can transfer more data than snapshot while maintaining the same failure rate. In other words, it offers more capability.

While performance is important for the DR process [11-12, 16-17], identifying the failure rates is critical to understand how effectively the DR process can take on large volumes of data over the network in an emergency [12-13]. All experiments should investigate the failure rates in losing or corrupting the data during and after the DR process. In order to validate our proposal with this multi-purpose approach, three major experiments are required as follows.

The first major experiment presented in Section 4.1 and 4.2 is focused on the data migration from the Big Data backup servers to emergency backup servers as the baseline. It also provides details of the performance and failure rate of losing/damaging files in migrating data between 1,000 files (totaling 1 TB) and 10,000 files (totaling 10 TB), and also large single files of between 100 GB and 1 TB.

The second major experiment shown in Section 4.3 is focused on snapshot recovery, a faster method to recover data, with results in performance discussed. Snapshot has a different approach for failures, and it starts with number of failed attempts, the data of which is presented.

The third major experiment is focused on replication to show its performance in the DR process, and investigation of failure rate. In order to compare the performance and failure rates, all results are put together for the comparison.

The hardware involved included a desktop working as a client, which had 3.40 GHz iCore 7, 16 GB DDR3 memory, 4 TB of SATA hard-disk. Network speed is 10 Gbps at the off-peak during the data recovery took place. All the VMs used in ULCC in London, sites in Southampton and Leeds have the same hardware specifications to ensure a fair comparison between all hardware and VM specifications. If the required hard disk will need more than 3.6 TB (to ensure the VM is not down, that is the recommended maximum capacity), then backup files will be transferred to the next VMs as illustrated in Figure 1 and Figure 2.

4.1 Experiments involved with data transfer from the Big Data servers to emergency backup servers

This set of experiments is involved with transferring data from the Big Data backup servers to the emergency servers without fault tolerance, and is focused on the transfer of data, rather than snapshots with VMs, or replications to the mirror sites. Fault tolerance can be used but in an emergency, it is optional since it may prolong the time for rescuing critical data. Hence, experiments with and without fault tolerance were undertaken with key results recorded. Both the GSTT and ULCC have a fiber optic network with a maximum speed of 1.2 gigabytes per seconds (GBps), or 9.6 gigabits per second. Since data is measured in terms of bytes, the network speed used for this paper is measured in bytes. Before the experiment began, the network speed was measured over a period of time to obtain the best average network speeds, which were 400 MBps (megabytes per second) for upload speed and 700 MBps for download speed. The transfer of data is considered to use upload speed, since data is 'uploaded' onto the server, and 'sent across' the network. For the purpose of the experiment, each file is 1 GB in size, so the experiments were involved with sending between 1 TB (1,000 files) and 10 TB (10,000 files) of data across network. It means the expected execution time is 1000 / 0.4 = 2500 seconds = 41 minutes and 40 seconds for sending 1 TB of data from the Big Data servers to the emergency backup server if all data backup process went smoothly, that is without failures.

4.1.1 DATA MIGRATION BETWEEN 1,000 AND 10,000 FILES

The experiment performs automated backup of between 1,000 and 10,000 files, which are available in the existing system for user support. 1,000 of medical record files have the total size of 1 TB and 10,000 files of medical record files have the total size of 10 TB. Each set of the experiments is performed three times with the average time obtained, where variations in time taken are also presented in the graph shown in Figure 3. All the data migration can be completed below 25,000 seconds for up to 10 TB data migration. The variations in the total execution time are kept below 3% of each other for all experiments. The graph shows a straight line, meaning that the execution time rises in proportion to the size of data transfer. The difference between a desktop and the Cloud in all the experiments in Section 4 is that a desktop will stop at 4TB hard disk limit if not using Cloud, whereas the Cloud can ensure business continuity by backing up files in different VMs.

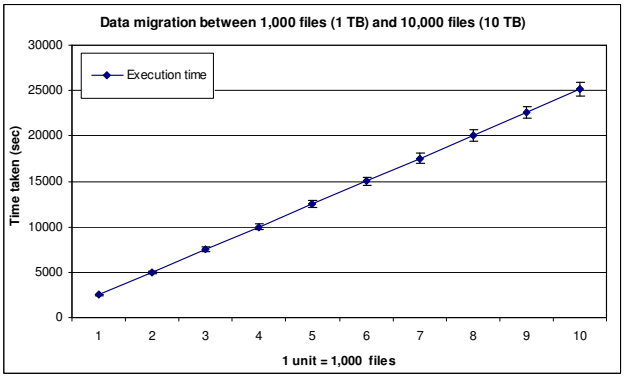

Figure 3: Data migration for 1,000 files (1 TB) to 10,000 files (10 TB)

4.1.2 DATA MIGRATION OF SINGLE LARGE FILES

Data migration is common amongst clouds and is also relevant to the DR process. When there are more organizations going for Private Cloud deployment, data migration between clouds is common and may influence the service delivery [2, 6, 9, 18, 20]. After the end of medical experiments and clinical activities, images from tumors and patients have to be scanned, analyzed and stored. As a result, a large size of data can be generated. The current medical practice is all these images are stored as a single zip file, so that they can be archived according to the medical record and date rather than sending off hundreds of files altogether. Each zip file is arranged in the unit of 100 GB up to 1 TB in size and transferring data of these sizes through backup is a challenge. It becomes increasingly important to investigate the impact of moving single large files between Private Clouds. Hence, the objective here is to identify the execution time for moving single large file. Each file is approximately 100 GB on average. It means it should take 100 / 0.4 = 250 seconds = 4 minutes and 10 seconds. Figure 4 shows all the results, which have straight lines for transferring single data between 100 GB and 1 TB. The next experiment included the data migration test for a single file (in .zip) of size between 1 TB and 10 TB. All experiments were measured three times to get the average values as shown in Figure 5. Instead of the multiplication of 10 due to ten times the size, the execution time varied between 10 to 25 times more. To investigate this further, the first-time failure rate was identified. It means that the migration was unsuccessful and the automated system then attempted to do it again. Hence, there was a much higher execution time. Fault tolerance is required to ensure that this does not happen. The first-time failure rate will be investigated and presented in Section 4.3.

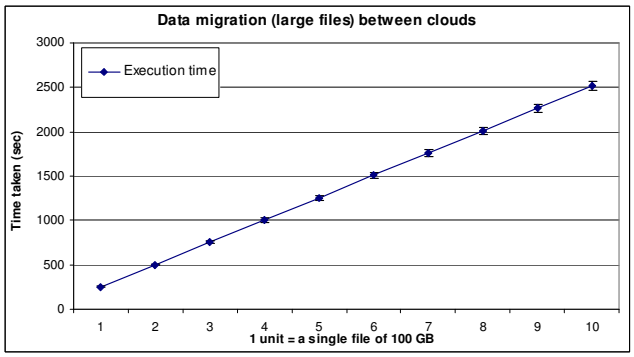

Figure 4: Data migration for a single file between 100 GB and 1 TB

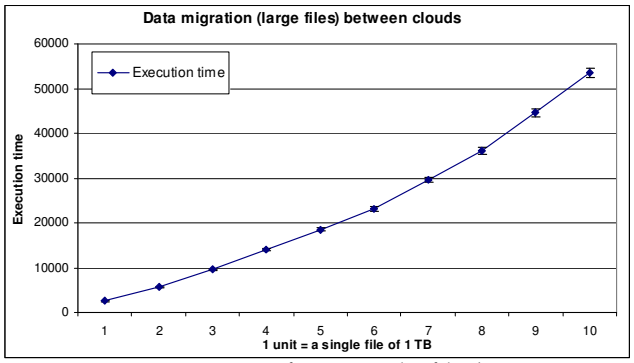

Figure 5: Data migration for a single file between 1 TB and 10 TB

4.2 FAULT TOLERANCE

Fault tolerance is an important step to ensure that the DR process can go ahead smoothly without being interrupted by

the failure rates. When the failure rates arrive to a certain level due to the difficulty of backing up single, large terabytes of data, DR process can manage the transfer, backup and recovery of files and data. In this section, we describe the method for the fault tolerance and the related experiments. The important message is to ensure that the jobs can be completed without interrupting the entire process. However, the limitation is that the entire process to backup and recover data can last longer. We present our algorithms to demonstrate the fault tolerance. The term risk is a variable used to define the percentage of failure rate. The Big Data system first reads the status of the jobs, and check the number of completed and failed jobs. The failure rate (risk) can be calculated by failed jobs divided by completed jobs. If the failure rate is kept under 5%, the DR process can be completed without terminating the whole process. However, if the failure rate goes beyond the recommended 5%, the Big Data DR process will be stopped by the system which will also receive a report of incidence. See Table 8b for details.

Table 8b: The code syntax to initiate replication process

```
read(status(job))
read(complete(job))
read(failed(job))
risk = failed(job)) / complete(job)
If (risk <= 0.05)
  continue(status(job));
else
  stop(status(job));
  report(status(job));  // report that there is an error
  exit
end
```

4.3 THE PERCENTAGE OF FIRST TIME FAILURE RATES FOR DATA MIGRATION

The percentage first time failure rate in the DR process is important as each failure in service will result in loss of time, profit and resources. First time failure rate also means that if the failed jobs are removed and new jobs are started again in another run of service, the majority can be successfully completed. However, it is not time effective since all the jobs are designed to run and be expected for completion without more reruns and tests. This part of experiment is to calculate the percentage of failures, while the experiments shown in Figure 2 and Figure 3 are running in real-time and records the number of successful and failed operations. Failed operations happen in the DR process. Monitoring the failure rate is important as failures contribute to the development of risks.

4.3.1 THE FIRST TIME FAILURE RATE IN DATA MIGRATION BETWEEN 1,000 AND 10,000 FILES WITH AND WITHOUT FAULT TOLERANCE

This section describes failure rate results in data migration of between 1,000 and 10,000 files between Big Data backup server and emergency backup servers with and without fault tolerance. Results shown in Figure 6 confirm the case. Failure rate is the percentage of data transfers between different Big Data system servers that is either incomplete or not successful. The failure rate is independent of the data migration in the DR process. In other words, if a large file cannot be transferred and backed up to another site, the Big Data system can keep trying it again until the previous failed job is completed.

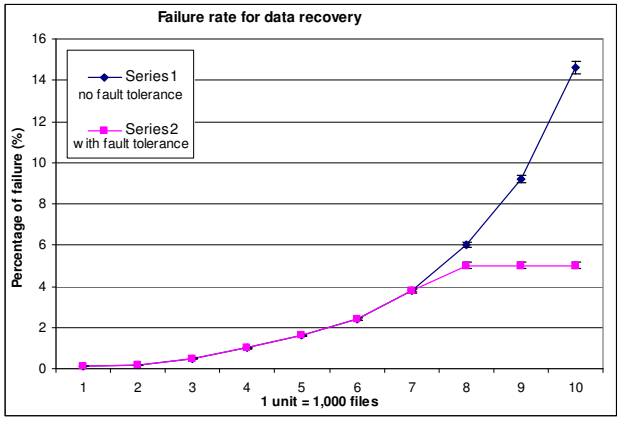

Figure 6: Failure rate of data migration between 1,000 files (1 TB) to 10,000 files (10 TB)

The entire test period requires three years with two rationales as follows. First, we ensure that we have similar and reproducible results in all experiments. Second, validity, accuracy and reliability need a significant period of time to verify in our previous studies [10, 23]. This shows that as the total data size increased, the percentage of first time fail-

ure rate increased, from 0.3% for 1,000 files (1TB) to 14.9% for 10,000 files (10 TB). Although the failure rate with fault tolerance can be maintained at 5%, it took more time in the DR process due to the two reasons. The first reason is that the Big Data system moves the failed job aside and continues to the next job without interrupting the entire DR process. The system then records the status of all the job requests. When all jobs are completed except the failed jobs, the Big Data system continues to process the failed jobs until they are successful. The second reason is that the DR recovery with fault tolerance cannot handle with large single files as presented in the next section.

4.3.2 THE FIRST TIME FAILURE RATE IN DATA MIGRATION OF A LARGE SINGLE FILE WITH AND WITHOUT FAULT TOLERANCE

Data migration of large files in the Cloud is common and important as storage is designed for terabytes and petabytes. The failure rate is shown in Figure 7 based on the number of successful and failed operations since 2009. Similar to Figure 6, the curve is close to an exponential one without fault tolerance, which means the failure rate increases significantly as the size of the migrated file increases. The rate of failure is higher than Figure 6. It started from 0.3% for one 100 GB file to 20.4% for one 1 TB file. Similar to Figure 6, the curve is close to an exponential one, which means the failure rate increases significantly as the size of the migrated file increases. The rate of failure is higher than Figure 6. It started from 0.3% for one 100 GB file to 20.4% for one 1 TB file. However, the major difference is that data migration with the fault tolerance cannot cope with the large single files of the size above 800 GB per file. The Big Data system would be terminated for the DR process if each file size is above 800 GB. This means that during the emergency status such as fire, the DR process should be in favor of without fault tolerance if the Big Data system has bulky single files that each one is above 800 GB in size. Fault tolerance can be applied by using another model such as Organizational Sustainability Modeling (OSM) [24]. If OSM is set 5%, it means the DR process can be temporarily suspended when it reaches 5% of the first time failure rate as shown in Figure 7. OSM will choose another method or another location to continue the DR process, so that continuity is always available to ensure more data can be backed up and recovered on time.

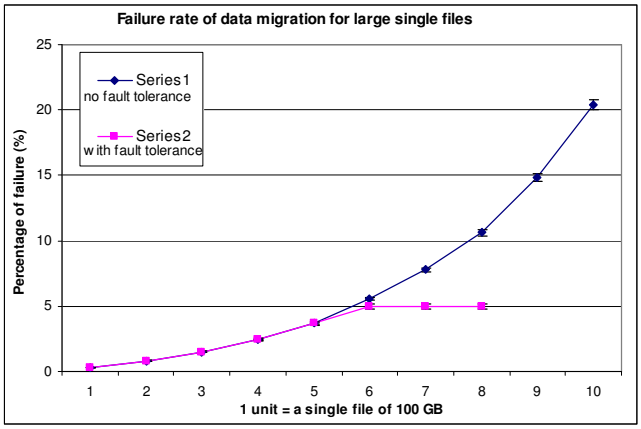

Figure 7: Failure rate of data migration for large single files (100 GB to 1 TB single file)

4.4 EXPERIMENTS WITH SNAPSHOTS

Snapshots are features associated with VMs. A single snapshot command or click can restore to the VM status when the snapshot was taken. This makes recovery much easier. Two types of snapshots are used. The first type is the snapshot offered by the VMware. The advantage is that it is easier to use and manage but the limitation is that it does not offer 100% recovery and the recovery performance is subject to a few factors. The second type is the snapshot offered by GlusterFS cluster, which is available in Linux distributions used for disaster recovery, particularly the recovery of the snapshots. The advantage is that it has a reliable rate of snapshot recovery closest to 100%, but the limitation is that it only deals with libraries and main files of the operating systems and not the entire organizational files. Two types of snapshots are used to ensure that all the files and data can be rescued and recovered.

A snapshot of each VM was taken at 7 pm each day. The time taken is dependent on the size of the VM, and the size of the file(s) that each VM contains. For example, a 8GB VM with no files inside it, takes less time for snapshot recovery than a 8GB VM file which is also using 5GB of files. For the purpose of this experiment, there are different sizes of VMs ranging from 100 GB to 1 TB, though each VM contains the same number and size of files for backup. This experiment is focused on the performance of snapshot recovery of these VM images.

4.4.1 SNAPSHOT RECOVERY IN DATA MIGRATION BETWEEN 1,000 AND 10,000 FILES

Figure 8 shows performance for snapshot recovery by VMware and GlusterFS cluster. Snapshot recovery by VMware took an average of 200 seconds faster than the GlusterFS cluster and has a relatively better performance with results in Figure 7 for snapshot recovery of each VM between 100 GB and 1 TB. The time taken to complete the snapshot recov-

ery is directly proportional to the size of the VM. For a 1TB VM, it needs 1272 seconds for VMware and 1455 seconds for GlusterFS. As the size of the VM for snapshot recovery increased, the variability in execution time became higher. This was because the variability in the time to recover VMs is larger.

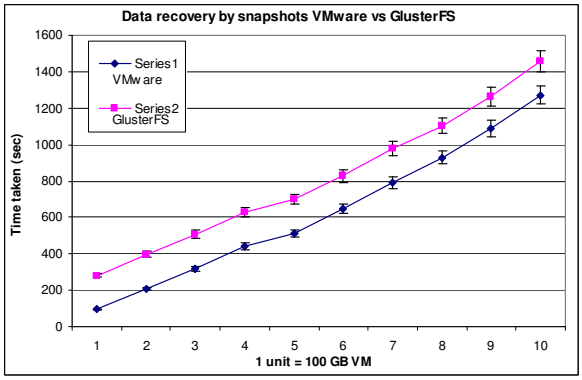

Figure 8: Data migration for a single VM between 100 GB and 1 TB between VMware and GlusterFS

4.4.2 NUMBER OF FAILED ATTEMPTS FOR SNAPSHOT RECOVERY

Snapshots are different from standard data transfer by TCP/IP protocols. Some snapshots can fail completely due to the complexity in the Big Data system and dependency between files. All snapshot recoveries are expected to be fully functional after a minimum of one attempt. Another set of experiments was undertaken to measure the average number of failed attempts to recover VMs by snapshots for VMware and GlusterFS, a process known as failover as described in Section 3.6. All results were taken from 2009 onwards. Results in Figure 9 show that the number of failed attempts for VMware and GlsuterFS follows an exponential curve. Failed attempts are still as low as 25 for up to 500 GB VMs, and then increase rapidly after this, until it reaches 151 times for recovery of 1 TB VMs. All the snapshots can be performed successfully without problems after the failed attempts, which can occasionally happen due to different variations and sizes in the data, as well as the health of network.

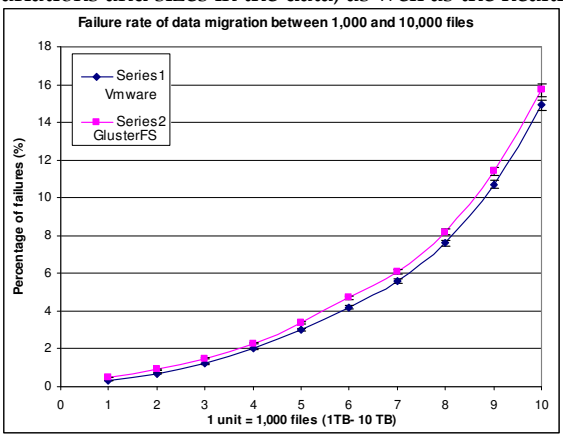

Figure 9: Number of failed attempts by snapshots

4.5 EXPERIMENTS WITH REPLICATION – A HYBRID APPROACH OF COMBINING TCP/IP AND SNAPSHOT

Section 3.4 describes the replications for data and VMs at mirror sites, and explains the difference between the snapshot recovery and replication, the latter of which takes more time than snapshot since system data and VMs are included. To experiment with the performance of replication, system data is limited to 100 GB. This means that replication requires recovery of VMs (which can be done by snapshot) and TCP/IP method (for transfer system data), and is the hybrid solution of combining both methods. The mirror site is about 50 meters away from the actual Big Data servers, and has the same network speed of 400 MBps. When sending backup data of 100 GB across the network, this implies additional time of 100/0.4 = 250 seconds. Results are presented in Figure 8, which shows a straight line and all DR process can be completed less than 1,600 seconds. Replication is quicker than when using the TCP/IP method but slightly longer than snapshot.

Section 4.3.3 reported that the number of failed snapshot attempts increased when the size for snapshot recovery increased. Replication also has the same results, as shown in Figure 9. The next step is to investigate the percentage of lost and damaged files in replication. Results will be discussed in Section 4.5.

Although replication uses the hybrid approach, it can pose a greater challenge for system administration, as more scripts or enterprise solutions are needed if the DR process is to offer both snapshot recovery and emergency backup

on TCP/IP. If the system architecture and programming model for replication is not robust, the DR process may not be completed successfully, because either snapshot recovery is done without the system data transferred or vice versa; or the DR process arrives at the stop function without being able to resume.

4.6 COMPARISONS OF RESULTS BETWEEN TCP/IP BASELINE, SNAPSHOT AND REPLICATION

Two types of comparisons between TCP/IP baseline, snapshot and replication are presented in this section. The first comparison is focused on the performance experiment (time taken) and the second comparison is focused on the experiment showing the failure rate of losing or damaging data during the DR process. For the purpose of comparison, the TCP/IP adopts no fault tolerance to allow data backup of large single files and snapshot uses VMware as the default due to the better performances over the GlusterFS.

4.6.1 PERFORMANCE COMPARISON

This experiment uses 100 to 1,000 files (100 GB to 1 TB) to compare the performance by TCP/IP baseline, snapshot and replication. Set up is the same as described in the earlier portion of Section 4. Figure 10, in which each point is the mean of five runs of the experiment, shows that the TCP/IP baseline needed the highest execution time of 2,500 seconds to complete up to 1,000 files (1 TB) of data transfer in the DR process. Both snapshots and replications only required half of the execution time of TCP/IP baseline.

4.6.2 COMPARISON IN FAILURE RATES

The next experiment is involved with checking the data consistency to identify whether any data has been lost, damaged or corrupted in the DR process. To make substantial comparison, a large quantity of files is more suitable. However, it cannot contain too many files and is limited up to 5,000 files, since it will take longer time to inspect, and also the VM image will be in terabytes and harder to move across network shown in Figure 3 and Figure 4.

The case between 1,000 files (1 TB) and 5,000 files (5TB) are used in the TCP/IP baseline, VM snapshot and replication. Results are taken five times with variations kept in 2%, and are shown in Figure 11.

Our results show that while the replication is only 1% better than the snapshot recovery, both snapshot and replications are quicker than the TCP/IP baseline. The tradeoff for better performance is that it has a higher percentage of lost and damaged files during the DR process. The percentage went up to 5% of incurring lost, damaged and corrupted files. Our experiments show that more files are lost and damaged in the DR process.

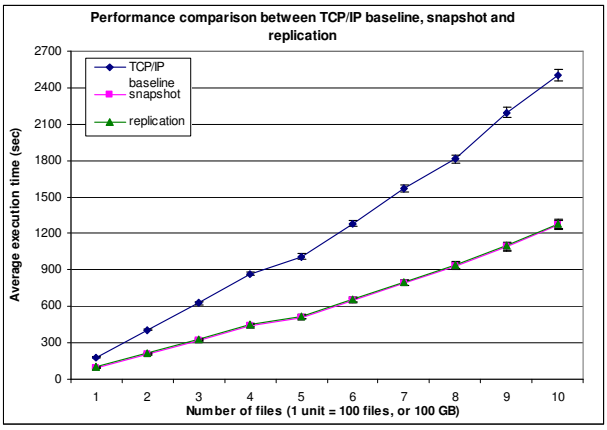

Figure 10: Data migration for a single file between 100 GB and 1 TB

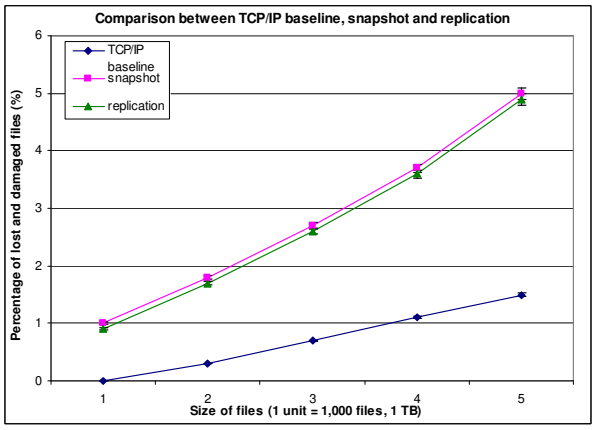

Figure 11: The average execution time taken between TCP/IP baseline, snapshot and replication

Data is kept safe and secure in our proposed solution. The results in Section 4.4 show that TCP/IP baseline method is more reliable for securing a high percentage of data but takes more time, whereas snapshot and replication are both quicker at the expense of some reliability. There are two approaches to achieve both speed and data consistency. First, all the methods are used at the same time, so that results can complement with each other. If the organization needs restore data faster, they can retrieve it from either snapshot or replication. If the organization's priority is accuracy and reliability in data preservation, then they can use TCP/IP method. Second, more snapshots can be used. This ensures that better performance can be achieved. Where less than 5% of data is lost or corrupted, some of these data can be recovered in other VMs. Advanced intelligent systems can be used to identify lost and damaged files, and then recover them from tape, other backup servers or other backup images.

4.7 CHECKING THE PERCENTAGE OF FILES RESTORED IN THE FULL RECOVERY IN A HAZARDOUS ACCIDENT

This section presents investigations of the percentage of files in the full recovery based by the combined approaches shown in Section 4.1 and 4.2 (TCP/IP method, baseline), 4.3 (snapshot) and 4.4 (replication). This experiment was

conducted before announcing the offering of Private Cloud services to medical users in order to ensure the disaster recovery is fully functional. DR test results allow the stakeholders to understand the extent of full recovery, and whether the DR process can recover data in high percentages. The results are based on

1. Checking the number of files before and after the DR process.
2. Checking the disk space used by all the backed files before and after the DR process.

Table 8c: Full recovery test in a non-emergency

Methods	Number of recovered files (out of 10,000)	Notes
TCP/IP baseline	9838	May not be recommended if fire spreads quickly.
Snapshot	9499	Quicker but fewer recovered files than TCP/IP.
Replication	9501	Same as above.
All three (multi-purpose; no duplicated files)	9994	Some missing files were found in other VMs

For the first set of checks, the number of files before the DR process was 10,000. After the DR process, the number of files backed up and recovered fully was 9,994, corresponding to 99.94% of data recovery. For the second check, the disk space after the DR process was 99.9% of the disk space before the DR process. Results for the first check are recorded in Table 8b.

Results in Table 8c show all the three methods could recover between 95% and 98.4% of the total number of files. The combined effort of the three methods had 9,994 files with 99.94% recovery. The reason is that "multi-purpose" approach allows that each VM to restore about 95% of the data, though some are not the same. As a result, the total recovery can go up to 99.4%. Although TCP/IP has a high percentage of full recovery, it takes much more time, and may not be recommended if the hazard is out of control, such as fire is spreading quickly. Results in the real accident showed that the proposed DR approach can recover a substantially high percentage of data. Additionally, checking the content of files before and after the DR process will be undertaken, since it is harder to check all the content and verify that there is no any change.

4.8 SUMMARY OF ALL EXPERIMENTS

Section 4 presents all the experiments associated with the DR process, and discusses results for each section. Results in the first experiment include the following:

1. Data migration between 1,000 and 10,000 files via TCP/IP: 10 TB of data can be transferred in 25,000 seconds (less than 7 hours).
2. Data migration of large single files: Up to 1 TB of large single files can be transferred in 2,500 seconds (less than 42 minutes).

The first experiment began to display an exponential curve after migrating 5,000 files (500 GB), and increased from 0.3% for 1,000 files to 14.9% for 10,000 files. The second experiment had the similar behavior except having higher failure rate, and started from 0.3% for one 100 GB file to 20.4% for one 1TB file.

The second major experiment concerned taking snapshots of VMs. It had a shorter time taken than the TCP/IP method, and took less than 1,300 seconds (21 minutes and 40 seconds) to recover 1,000 files, or 1 TB of data. However, the number of failed attempts to take snapshot recovery increased while the size of the VMs became larger, which looked like an exponential curve.

A proposed solution is to use a hybrid approach of replication, which has features of both TCP/IP baseline and snapshot. Replication from mirror sites takes 99.9% of the execution time of the snapshot. The percentage of lost and damaged files was 99.5% close to the results in snapshot recovery.

5 Comparisons between single and multi purpose approaches

This section describes experiments between single basket and multiple purpose approaches. Section 5.1 and 5.2 describe experiments in a non-emergency and an emergency respectively, whereby the difference between the single and multiple basket approaches are more noticeable in an emergency.

5.1 EXPERIMENTS FOR THE MULTI-PURPOSE APPROACH

Our contribution includes the design and deployment of the multi-purpose approach, so that all backup can be done in one single attempt with all sites received backup files by all the three methods. This ensures all valuable data can be kept safely with the minimum amount of loss.

5.1.1 COMPARATIVE EXPERIMENTS AT THE ULCC

Our solution offers a "multi-purpose" approach, which does not rely entirely on one technique, or one site as other

papers suggest. This enables that data can be restored from more than one location, ensuring that business continuity can be achieved and impacts to work efficiency can be minimized. The combined use of the three can save time but similarly, the first-time failure rate may rise with the increase in the data size. This section presents the experiments undertaken at the ULCC by using the state-of-the-art API, restore() function. This can save more efforts than conducting each method manually and thus provides a better efficiency since more data can be backed up and restored at all other sites at the same time, or use three methods at chosen sites(s) at the same time. The function restore(optimize) and restore(London) were used for experiments and results were taken three times for records. The function restore(optimize) can accelerate the speed of data recovery twice by using accelerated snapshot technique, however, the trade-off is that it can introduce higher first-time failure rate. The function restore(London) can use TCP/IP, snapshot and replication method. In the event of the fire, the fastest method is chosen by default, which is snapshot. This is a fairer comparison since it is comparing the same method. Results in Figure 12 show that the execution time to complete data recovery of restore(optimize) is below 650 seconds to recover 1 TB of data and is 50% of the time completed by restore(London).

While data recovery completion was considered fast (under 650 seconds for optimize function). Percentage of failure rates should be identified since it is undesirable to lose data in the process of data recovery which can have impacts on businesses. Performance measurement for experiments in Figure 12 was conducted three times. The percentage of failure rate can be calculated by the number of files failed to be recovered versus the number of files successfully recovered. If a 100 GB contains 10,000 files and 20 were not recovered, the percentage of failure rate was 20 / 10000 = 0.2% for 100 GB. All the measurements were recorded. Figure 13 shows results of comparison between restore(optimize) and between restore(London), whereby the rate of failure in restore(optimize) is higher. There was zero percent of failure rate at 100 GB for restore(London) and 200 100 GB for restore(optimize). The highest comparison is between 1.6% for restore(optimize) and 1.0% for restore(London). Hence in the event of fire, restore(optimize) is still worth to go ahead due to the low percentage for 1 TB and has half the execution time of the other API command.

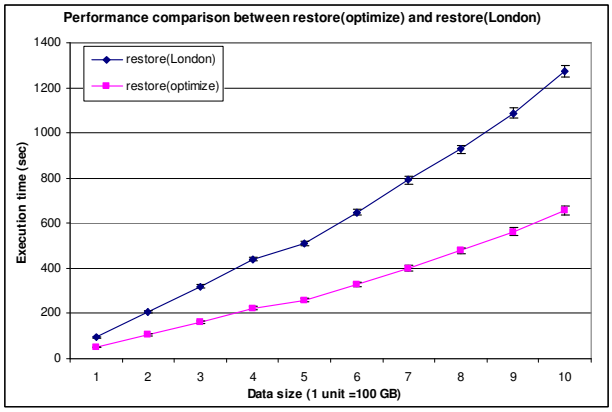

Figure 12: Performance comparison between restore(optimize) and restore(London)

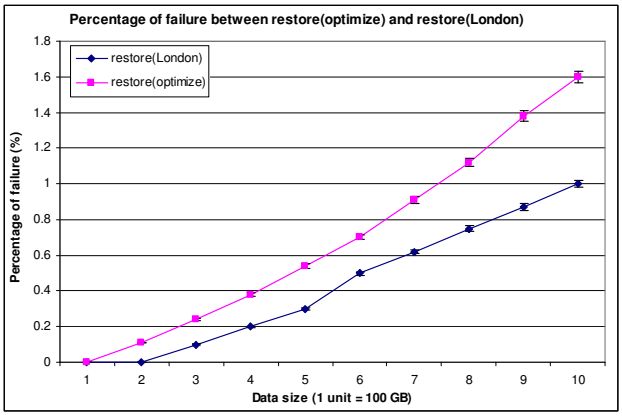

Figure 13: Percentage of failure rate between restore(optimize) and restore(London)

5.1.2 Experiments between the ULCC, Southampton and Leeds

This section describes experiments between the ULCC, Southampton and Leeds as part of the multi-purpose approach. All the experiments were focused on the snapshot since the quality of service for a full backup by the TCP/IP method due to the longer distance involved. Snapshot recovery is reliable and efficient. The restore() function was adopted whereby restore(default) was used since it could serve and restore at all sites at the ULCC, Southampton and Leeds. Experiments undertaken in Section 5.1.1 were then repeated for this set of experiments which also involved restore(Southampton) and restore(Leeds). Similarly, in the event of accidents such as fire, snapshot will be chosen as the default method at each location and is then executed first.

Results in Figure 14 show the execution time to use restore(London), restore(Southampton) and restore(Leeds) for data recovery and all the tasks can be completed within 1,400 seconds. All the execution time was within 3% of each other, with restore(London) the fastest, followed by restore(Southampton) and restore(Leeds). A likely reason is due to the distance, since the distance between the default site in London is the nearest to the DR site in ULCC and the network does not need to travel a longer distance, which also explains why restore(Leeds) can take longer than the other two commands. Figure 15 shows percentage failure comparison between restore(London) the fastest, followed by restore(Southampton) and restore(Leeds). Similarly, restore(London) has the lowest of 1% of failure rate for data up to 1 TB. A shorter distance of travel can ensure a lower failure rate. The command restore(Southampton) has the highest failure rate of 1.4% and restore(Leeds) has 1.2% for data size up to 1 TB.

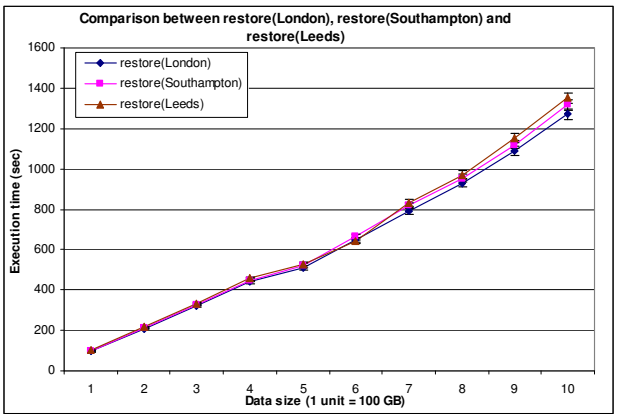

Figure 14: Performance comparison between restore(London), restore(Southampton) and restore(Leeds)

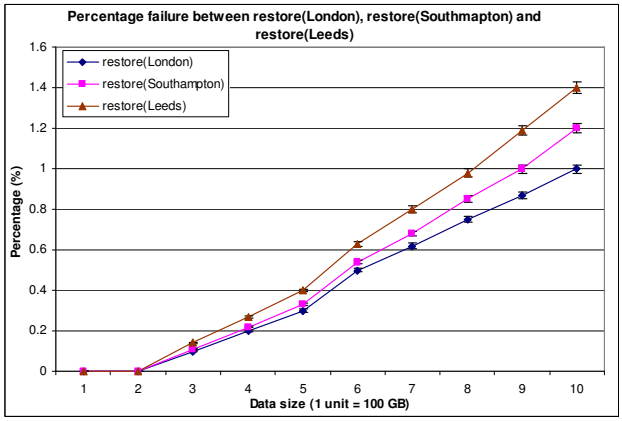

Figure 15: Percentage of failure rate between restore(London), restore(Southampton) and restore(Leeds)

5.2 MULTI VERSUS SINGLE BASKET APPROACH IN AN EMERGENCY

There was a real accident happened in February 2009 due to a high temperature hazard causing hard disk failures and even a possibility of fire, in which DR process was initiated, taking hours to complete the entire process. Results in this section were based in a real accident, since it was a real data and provided a suitable 'testbed', rather than experiments to show the technical capacity. In the event of emergency, backup speed requires the quickest time to be completed and does not rely on TCP/IP method which can be prone to the event of fire (if network cables and infrastructure are at risk). Results are shown in Table 9. Three single basket approaches obtain 0, 9383 and 9011 data for recovery. On the other hand, the multi-purpose approach restores all of 10,000 data successfully if considering all the sites in London, Southampton and Leeds. With the default method, the multi-purpose method can recover all the data. Since the default location is ULCC in London, the number of recovered data has been recorded. Default method had 9,987 data recovered and the optimize method in multi-purpose DR had 9,905 data recovered. It means that although the optimize method can accelerate the speed of backup, the tradeoff is that some data cannot be fully recovered to the default venue. Collectively all three sites can recover all 10,000 data. The TCP/IP baseline is the most adopted method in DR process and our results show that in the event of fire, it is not a reliable method and a combination of different methods of providing DR services should be deployed.

Table 9: Full recovery test in an emergency

Methods	Number of recovered files (out of 10,000)	Notes
TCP/IP baseline	0	In the event of fire, this method is the least reliable.
Snapshot	9,383	Recover a high extent of data.
Replication	9,011	Fewer than snapshot because some data rely on TCP/IP.
Multi-purpose (default)	10,000 (all) and 9,987 (London, default venue)	Some missing files were found in other VMs
Multi-purpose (optimize)	10,000 (all) and 9,905 (London, default venue)	Faster but not slightly fewer data to be recovered successfully

5.3 RECOMMENDATION

Comparisons between three methods were undertaken. Both snapshots and replications only required about half of the execution time required by the TCP/IP baseline method. In contrast, the TCP/IP baseline method had the better reliability and accuracy in data preservation. The tradeoff for the better performance was that it had a higher percentage of lost and damage files; up to 5% for 5TB compared with approximately 1.5% of 5TB data in the TCP/IP method. In the event of emergency, the quicker execution time to complete DR processes is more important than data preservation and only snapshot and replication are used. The multi-purpose approach can still maintain a high percentage of files to be recovered successfully in the event of an emergency. The results showed 99.05% full recovery for files, and 99.0% of disk space restored. The advantages of adopting multi-purpose approach are that it ensures the DR process to be done more quickly, efficiently and safely. In the accidents like fire, the DR process should back up all the files to as many sister sites as possible and cover as many methods as possible, in case that if some backup has failed or struggle to be completed.

Our work has two important aspects. First, there are not many papers describing DR processes. Most of Cloud computing papers only describe from the users and the experiments points of view, without including details about

how to run and manage data centers for hosting Private Cloud services. In industry where healthcare and finance, the sensitivity of data is important whereby the data is stored and protected safely in the Private Cloud. We demonstrate details and techniques required in the Private Cloud management to ensure that more than 99% of data can be restored and protected. Second, the emergence of Big Data poses a challenge to protect and recover the quantity, size and complexity of the data. We offer a case study (based on ULCC) approach to ensure that terabytes of data can be restored efficiently. Then the DR process and tests must be investigated to ensure that the restoration can provide a huge capacity. All the lessons learned including the adoption of multi-purpose approach for DR process can be useful for service providers or institutes that manage Big Data systems and applications.

5.4 How does the proposed solution meet criteria for Big Data

This section explains how our DR process can meet criteria for Big Data. First, the proposed solution meets the criteria for volume, since it handles with 10 petabytes of data at the ULCC Data Center and the experiments involved with 10 terabytes of data were backed up and recovered at multiple sites. Second, the proposed solution meets the criteria for velocity since it allows a rapid management of data. Third, the proposed solution meets the criteria for variety since there are different types and forms of data involved in the DR process. Fourth, the proposed solution meets the criteria for veracity since there is a extremely high percentage of data being fully recovered as illustrated by the experiments. Last, the proposed solution meets the criteria for value since data involved with medical and scientific research over a period of many years can be stored and recovered to allow scientists to work on the data even if the major accident such as fire had happened. Different sets of experiments were conducted to demonstrate that all data recovery can be completed. For example, it took less than 650 second for 1 TB of data for restore (optimize) and there was only 3% difference in execution time between restore(London), restore(Southampton) and restore(Leeds). The percentage of failure rate for all restore API commands are between 1% and 1.6%, showing that there is a good reliability in data backup and recovery for Big Data services.

6 Conclusion and future work

Our paper demonstrates a "multi-purpose" approach to ensure that restored data can be fully recovered from multiple sites, with three methods used. The system design and development for these three methods has been explained. The traditional TCP/IP baseline, snapshot of VMs and replication have been jointly used. This ensures that an extremely high percentage of data can be fully recovered; 99.94% full recovery of data in a hazardous event. Experiments of the three methods have been undertaken. The TCP/IP baseline is focused on reliability in preserving a higher percentage of data restored during the DR process. Snapshot is focused on better performance to recover rescued data. Two types of snapshot approaches are used with the VMware snapshot used as the default method for experiments. The hybrid approach of replication behaves more like snapshot. Hence, comparisons between TCP/IP baseline and snapshot/replication are made. We also discover that methods with fault tolerance can provide low failure rates of data migration but cannot cope with data migration of large single files above 800 GB in the DR process. Snapshot by VMware has slightly better performances than GlusterFS.

Experimental results show that snapshot/replication can complete full recovery twice as fast as the TCP/IP baseline, but has 5% data loss and damaged compared to 1.5% in the TCP/IP baseline. Comparisons between multi and single basket approaches are made whereby the multi-purpose approach shows a real difference in an event of emergency. Multi-purpose approach offers 99.05% recovery of all the 10,000 data. In terms of single basket approach, TCP/IP baseline (as the most adopted method) does not restore any. The use of multi-purpose approach should be adopted in data centers for hosting Private Clouds or Big Data services to ensure that a large quantity and volume of data can be restored as efficient as possible. This has made an important research contribution since many Big Data papers focus on the performance and do not even conduct experiments to test the DR process for the Big Data and investigate the results used by different methods. All our 10 petabytes of the data in the ULCC data center have been used for experiments over a period of two years to ensure the reliability of our results. Explanations about how our proposed work meets volume, velocity, variety, veracity and value have been explained.

We developed a state-of-the-art DR approach that backed up all the data to all sister sites in London, Southampton and Leeds involved with large scale experiments with real data and tests. The restore() API makes the data recovery and back up intelligently and efficiently. Results showed that multi-purpose approach could recover 1TB of data within 1,400 seconds for all sites and within 650 seconds for optimize method. Failure rates are kept between 1% and 1.6% of 1 TB of data during experiments. All these results contribute to the important decisions in the event of fire, since all data should be backed up and recovered as soon as possible and businesses do not lose data. Our future work will include integrations with existing data centers and perform large-scale data recovery on our upgraded Private Cloud which will provide services for petabytes of data sending and backing up across our private networks on the Private Cloud and ensure all biomedical scientists will be able to perform Big Data backup, recovery and computational services on the Cloud. Future work also develops intelligent systems to decide which method to action and self-minimize the first-time failure rates.

References

[1] J., Gantz, D., Reinsel, Extracting value from chaos. IDC iView, 2011, 1-12.

[2] M., Chen, S., Mao, Y., Liu, Big Data: A Survey. Mobile Networks and Applications, 19(2), (2014), 171-209.

[3] C. L. P., Chen, C. Y. Zhang, Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Information Sciences, 275, (2014), 314-347.

[4] D., Agrawal, S., Das, A., El Abbadi, (2011, March). Big data and cloud computing: current state and future opportunities. In Proceedings of the 14th ACM International Conference on Extending Database Technology, March 2011, 530-533.

[5] H., Chen, R. H., Chiang, V. C., Storey, Business Intelligence and Analytics: From Big Data to Big Impact. MIS quarterly, 36(4), (2012)., 1165-1188.

[6] M., Armbrust, A., Fox, R., Griffith, A. D., Joseph, R. , Katz, A., Konwinski, G., Lee, D., Patterson A., Rabkin, I., Stoica, M., Zaharia, A view of cloud computing. Communications of the ACM, 53(4), 2010, 50-58.

[7] P. T., Jaeger, J. , Lin, J. M., Grimes, Cloud computing and information policy: Computing in a policy cloud?, Journal of Information Technology and Politics, 5(3), (2008), 269-283.

[8] V., Kundra, Federal cloud computing strategy, US Government white paper, Feb 2011.

[9] A., Khajeh-Hosseini, D., Greenwood, I. Sommerville, Cloud migration: A Case Study of Migrating an Enterprise IT System to IaaS. 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), July 2010.

[10] V., Chang, R. J., Walters, G., Wills, Cloud Storage and Bioinformatics in a private cloud deployment: Lessons for Data Intensive research. In Cloud Computing and Service Science, Springer Lecture Notes Series, Springer Book, 2013.

[11] A., Meissner, T., Luckenbach, T., Risse, T., Kirste, H., Kirchner, Design challenges for an integrated disaster management communication and information system. In the First IEEE Workshop on Disaster Recovery Networks, Vol. 24, June, 2012.

[12] A. S., Szalay, G. Bell, J., Vandenberg, A., Wonders, R., Burns, D., Fay, J., Heasley, T., Hey, M., Nieto-SantiSteban, A., Thakar, C., van Ingen, R., Wilton, Graywulf: Scalable clustered architecture for data intensive computing. In the 42nd Hawaii IEEE International Conference on System Sciences, HICSS'09, Jan 2009, 1-10.

[13] J., Elerath, Hard-disk drives: the good, the bad, and the ugly. Communications of the ACM, 52 (6), June 2009.

[14] C. J., Hiatt, A Primer for Disaster Recovery Planning in an IT Environment, Idea Group Publishing. ISBN 1-878289-81-0, 2000.

[15] B. R., Kandukuri, V. R., Paturi, A., Rakshit, Cloud security issues. In IEEE International Conference on Services Computing, SCC'09, 2009, 517-520.

[16] M., Pokharel, S., Lee, J. S. Park, Disaster recovery for sys-tem architecture using cloud computing. In the 10th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT), July 2010, 304-307.

[17] T., Wood, E., Cecchet, K. K., Ramakrishnan, P., Shenoy, J., Van der Merwe, A., Venkataramani, Disaster recovery as a cloud service: Economic benefits & deployment challenges. In 2nd USENIX Workshop on Hot Topics in Cloud Computing, June, 2010.

[18] S., Subashini, V., Kavitha, A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, 34(1), (2011), 1-11.

[19] S., Snedaker, Business continuity and disaster recovery planning for IT professionals. Newnes, 2013.

[20] T., Wood, H. A., Lagar-Cavilla, K. K., Ramakrishnan, P., Shenoy, J. Van der Merwe, PipeCloud: using causality to overcome speed-of-light delays in cloud-based disaster recovery. In Proceedings of the 2nd ACM Symposium on Cloud Computing, Oct 2011, p. 17.

[21] S., Sengupta, K. M., Annervaz, Multi-site data distribution for disaster recovery—A planning framework. Future Generation Computer Systems, 41, (2014), 53-64.

[22] V. Chang, The business intelligence as a service in the cloud. Future Generation Computer Systems, 37, (2014), 512-534.

[23] V. Chang, Cloud Bioinformatics in a private cloud deployment, In Advancing Medical Practice through Technology: Applications for Healthcare Delivery, Management, and Quality, IGI Global, 2013.

[24] V. Chang, A proposed model to analyse risk and return for Cloud adoption. Lambert Academic Publishing, 2014.

Dr. Victor Chang was an IT Lead in one of the NHS Trusts, UK. He is working as a Senior Lecturer at Leeds Beckett University since September 2012. Within four years, he completed PhD (CS, Southampton) and PGCert (Higher Education, Fellow) part-time, whereby the distance between his research and work is hundreds of miles away. He demonstrates ten different types of Cloud Computing and Big Data services with an extensive experience. He has over 70 peered-review publications up-to-date. He is the Editor-in-Chiefs of International Journal of Organizational and Collective Intelligence (IJOCI) and Open Journal of Big Data (OJBD), and is the Editor of prestigious Future Generation Computer systems (FGCS). He is a reviewer of several well-known journals. He has an expert knowledge in different domains of IT acquiring 27 certifications with 97% on average. He is the founding chair of two international workshops, one of which has stepped up as an international conference on Internet of Things and Big Data. He also gives keynote and invited talks in the UK and abroad. With 15 years of IT experience, he is a very active practitioner and researcher in Cloud Computing and Big Data in England.