Computer and Information Science; Vol. 7, No. 4; 2014
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

# Disaster Recovery in Cloud Computing: A Survey

Mohammad Ali Khoshkholghi[1], Azizol Abdullah[1], Rohaya Latip[1], Shamala Subramaniam[1] & Mohamed Othman[1]

[1] University Putra Malaysia, Selangor, Malaysia

Correspondence: Mohammad Ali Khoshkholghi, University Putra Malaysia, Selangor, Malaysia. E-mail: Khosh.kholghi63@gmail.com

Received: May 17, 2014 Accepted: June 24, 2014 Online Published: September 3, 2014

doi:10.5539/cis.v7n4p39 URL: http://dx.doi.org/10.5539/cis.v7n4p39

**Abstract**

Disaster recovery is a persistent problem in IT platforms. This problem is more crucial in cloud computing, because Cloud Service Providers (CSPs) have to provide the services to their customers even if the data center is down, due to a disaster. In the past few years, researchers have shown interest to disaster recovery using cloud computing, and a considerable amount of literature has been published in this area. However, to the best of our knowledge, there is a lack of precise survey for detailed analysis of cloud-based disaster recovery. To fill this gap, this paper provides an extensive survey of disaster recovery concepts and research in the cloud environments. We present different taxonomy of disaster recovery mechanisms, main challenges and proposed solutions. We also describe the cloud-based disaster recovery platforms and identify open issues related to disaster recovery.

**Keywords:** cloud computing, disaster recovery, replication, backup, survey

## 1. Introduction

Cloud computing becomes more popular in large-scale computing day by day due to its ability to share globally distributed resources. Users can access to cloud-based services through Internet around the world. The biggest IT companies are developing their data centers in the five continents to support different cloud services. The total value of the global cloud computing services market revenues is expected to reach about $241 billion by the end of 2020 (Reid et al., 2011). Rapid development in cloud computing is motivating more industries to use variety of cloud services (Arean, 2013), for instance near to 61% of UK businesses are relying on some kinds of cloud services (White paper, 2013). However, many security challenges have been raised, such as risk management, trust and recovery mechanisms which should be taken into account to provide business continuity and better user satisfaction.

Disasters, either manmade or natural, can lead to expensive service disruption. Two different disaster recovery(DR) models can be used to prevent failure in a network or CSPs : Traditional and cloud-based service models. Traditional model can be used as either dedicated infrastructure or shared approach. Based on speed and cost, customers can choose the appropriate model. In dedicated approach, an infrastructure is assigned to one customer, so both cost and speed is high. On the other hand, in the shared model (we can also call it distributed approach) an infrastructure is assigned to more multiple users. This approach decreases both cost and speed of recovery. As shown in Figure 1, cloud computing is a way to gain both dedicated and shared model benefits. It can serve DR with low cost and high speed.

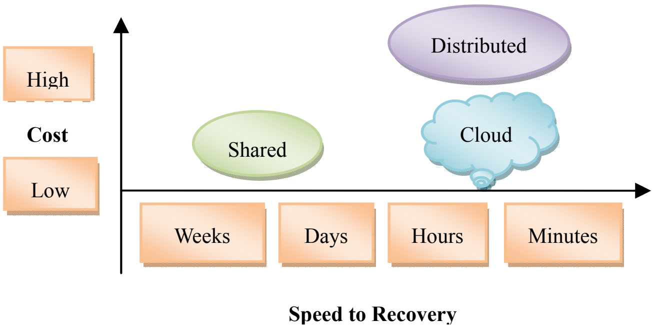

Figure 1. Comparison between traditional and cloud DR models (IBM White paper, 2012)

Table 1 shows a comparison between these three DR categories in terms of different features. Cloud computing decreases data synchronization between primary and backup site, minimizes different kinds of cost while increases independency between users' infrastructure and their DR systems.

Table 1. Disaster recovery models (Alhazmi and Malaiya, 2013)

| DR model | Data synchronization | Independency | Initial Cost | Ongoing cost | Cost of potential disasters |
|---|---|---|---|---|---|
| **Dedicated** | High | Low | High | Depends | High |
| **Distributed** | Medium | High | Medium | Depends | High |
| **Cloud** | Low | High | Low | Depends | Low |

According to IBM research (IBM white paper, 2012), only 50% of disasters in IBM are because of weather and the rest are because of other causes. For instance, such as cut power lines, server hardware failures and security breaches. Hence, DR is not only a mechanism for natural events, but also for all severe disruptions in cloud systems.

Organizations and businesses can use DR services which are served by cloud service providers. Using these services, data protection and service continuity are guaranteed for customers at different levels. Table 2 shows different DR services offered by IBM. In addition, one critical issue in DR mechanisms is that how can cloud providers tolerate disaster to prevent data lost and service disruption of their own data, infrastructure and services. In this paper we investigate both challenges and solutions for DR mechanism in cloud provider's point of view. For enterprises, the main goal of DR is business continuity which means resuming back services online after a disruption. Recovery time objective (RTO) and Recovery Point Objective (RPO) are two important parameters which all the recovery mechanisms try to improve. By minimizing RTO and RPO business continuity can be achieved. RTO is the time duration between disruption till restoration of service, and RPO denotes the amount of data lost after a disaster. Failover delays consist of 5 steps depending on the level of backup (Alhazmi and Malaiya, 2013):

S1: Hardware setup

S2: OS initiation time

S3: Application initiation time

S4: Data/process state restoration time

S5: IP switching time

Therefore, RPO and RTO can be defined as:

$$RPO \propto \frac{1}{\mathrm{Fb}} \tag{1}$$

Where Fb is Frequency of backup.

$$TO = fraction\ of\ RPO + \sum_{\mathrm{J\ min}}^{\mathrm{S5}} \mathrm{T_j} \tag{2}$$

Table 2. IBM different DR service level

| IBM SmartCloud recovery service level | Recovery time | Description |
|---|---|---|
| **Gold** | 1 minute | For mission-critical applications |
| **Silver** | 30 minutes | For rapid recovery |
| **Bronze** | 6 to 24 hours | Assisted failover and failback |

The rest of this paper is organized as follows: In the section 2 cloud computing has been introduced briefly. In the section 3 we discuss cloud-based DR in detail. In section 4 and section 5 we investigate main challenges in DR mechanisms and some proposed solutions, respectively. It is followed by section 6 discussing some cloud-based DR systems will be introduced. In the section 7, the Open issues have been investigated. Finally, the paper ends with the proposed overall DR procedure and conclusion.

## 2. Cloud Computing: A Brief Review

Cloud computing – a long held dream of computing as a utility – is a promising technique which shifts data and computational services from individual devices to distributed architectures. The content of cloud was initially created to describe sets of complex on-demand services offered by commercial providers (Armbrust et al., 2010; Buyya et al., 2010). Based on the advancement in network topology with high speed bandwidth and Smart phones, people can upload their information using the Internet anytime. Cloud computing denotes Internet-based distributed computing platforms which are highly scalable and flexible. Their features can change the fashion of conventional information processes. Cloud computing allocates IT resources, such as computational power, storage, software, hardware platforms and applications to a wide range of consumers, possessing a wide range of devices.

Cloud providers including -public, private or hybrid clouds- are able to offer seamless on-demand services as a pay-as-you-go model. Therefore, consumers can easily use the services without a need to install or worrying about the underlying infrastructure. So, they can focus on their applications and can scale and retrieve the allocated resources directly by interacting with Cloud Service Providers. Virtualization is the key enabling technology in which cloud computing can change the system's view from a piece of hardware to a dynamic and flexible entity (Salapura., 2012). Cloud-based services can be divided into three levels: Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Platform as a Service (PaaS) (Zhang., 2013).

According to the NIST (Mell., 2011; Liu., 2011), the essential features of cloud can be defined as: On demand self-service, broad network access, resource pooling, rapid elasticity, measured service. Taxonomy, advantages and challenges of cloud computing are shown in the Table 3 (Cloud taxonomy, online).

Table 3. Taxonomy, advantages and challenges of cloud computing

| | |
|---|---|
| **Taxonomy** | *Infrastructure services*: Storage, compute, services management.<br>*Cloud software*: Data, compute, appliances, file storage, cloud management.<br>*Platform services*: Database, business intelligence, Integration, development & testing.<br>*Software services*: Billing, financial, legal, sales, Desktop productivity, human resources, content management, backup & recovery, social networks, collaboration. |
| **Advantages** | Improved business continuity, On demand storage and compute power, lower cost of ownership, agility, pay as you go model, increased availability, mobility and collaboration. |
| **Challenges** | Security issues, Disaster recovery, Dependency, latency, transparency, performance concerns, SLA violations. |

## 3. Disaster Recovery

A disaster is an unexpected event in a system lifetime. It can be made by nature (like the tsunami and earthquake), hardware/software failures (e.g. , VMs' failure of Heroku hosted on Amazon EC2 on 2011) or even human (human error or sabotage). It can lead to serious financial loss or even can put human lives at risk (Kashiwazaki., 2012). Hence, between 2% and 4% of IT budget in huge companies is expended for DR every year (Prakash et al., 2012). Cloud-based DR solution is an increasing trend because of its ability to tolerate disasters and to achieve the reliability and availability. It can be even more useful in small and medium enterprises (SMEs), because they do not have much resources as big companies do. As shown in Table 4, Data level, system level and application level are three DR levels which are defined in terms of system requirements.

Table 4. DR levels

| DR level | Description |
|---|---|
| **Data level** | Security of application data |
| **System level** | Reducing recovery time as short as possible |
| **Application level** | Application continuity |

DR mechanisms must have five requirements for an efficient performance (Wood et al., 2010):

- Have to minimize RPO and RTO
- Have a minimal effect on the normal system operation
- Must be geographically separated
- Application must be restored to a consistent state
- Must guarantee privacy and confidentiality

### *3.1 Disaster Recovery Plan*

There are different DR approaches to develop a recovery plan in cloud system. They are based on the nature of the system. However in the literature, all these approaches are based on redundancy and backup strategies. The redundancy strategy uses separated parallel sites which have the ability to start up the applications after a disaster; whereas backup strategy uses replication technology (Lwin and Thein, 2009). The speed and protection degree of these approaches depend on the level of DR service that is shown in Table 5(Guster and Lee, 2011). In addition, three different types of replication technology are available: 1. Host and VM replication, 2. Database replication, 3. Storage replication.

Table 5. Cloud-based DR models

| **Model** | **Synchronize time** | **Recovery time** | **Backup characteristics** | **Tolerance support** |
|---|---|---|---|---|
| **Hot** | Seconds | Minutes | Physical mirroring | Very high |
| **Modified Hot** | Minutes | 1 hour | Virtual mirroring | High |
| **Warm** | Hours | 1-24 hours | Limited physical mirroring | Moderate |
| **Cold** | Days | More than 24 hours | Off site backup | Limited |

The objective of disaster recovery planning is to minimize RTO, RPO, cost, and latency by considering system constraints such as CPU, network and storage requirements. So we can say DR recovery planning can be considered as an optimization problem. According to (Nayak et al., 2010), DR plans include two necessary phases:

- Matching Phase: In this phase, all DR solutions have to be matched to the requirements of any data container (a data container means a data set with identical DR requirements)
- Plan composition phase: Selecting an optimal DR solution which can minimize cost with respect to required QoS for each data container.

ENDEAVOUR (Nayak et al., 2010) is a framework for DR planning process. As shown in Figure 2, it consists three modules:

- Input modules: Including DR requirements (such as protection type, RTO, RPO and application latency), Discovery engine (To find configuration information of primary and secondary sites) and knowledge repository (Replication technologies, instructions and composition formula).
- Planning Modules: Including solution generation (Analyzing DR requirements and matching them to replication techniques), Ranking (Sorting DR plans in terms of some attributes like cost, risk and latency (Azagury et al., 2002)) and Global optimization (selecting an optimal DR plan (Jaiswal et al., 2011)).
- Output: The output of ENDEAVOUR is an optimal DR plan for each application with some details like: target resources and devices, replication protocol configuration.

## 4. Disaster Recovery Challenges

In this section we investigate some common challenges of DR in cloud environments.

### *4.1 Dependency*

One of the disadvantages of cloud services is that customers do not have control of the system and their data. Data backup is on premises of service providers as well. This issue makes dependency on CSPs for customers (such as organizations) and also loss of data because of disaster will be a concern for customers. Dependency (Javaraiah, 2011) also creates another challenge which is the selection of a trusted service provider.

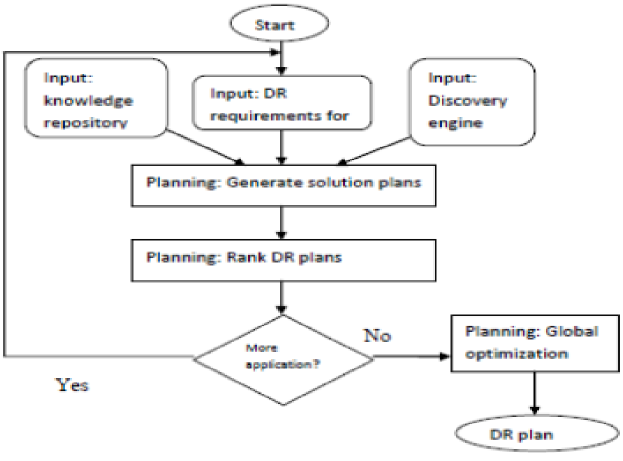
Start
Input: knowledge repository
Input: DR requirements for
Input: Discovery engine
Planning: Generate solution plans
Planning: Rank DR plans
More application?
No
Yes
Planning: Global optimization
DR plan

Figure 2. ENDEAVOUR flowchart

*4.2 Cost*

It is obvious that one of the main factors to choose cloud as a DR service is its lower price. So, cloud service providers always seek cheaper ways to provide recovery mechanisms by minimizing different types of cost. The yearly cost of DR systems can be divided in three categories (Alhazmi and Malaiya, 2012):

- Initializing cost: amortized annual cost
- Ongoing cost: storage cost, data transfer cost and processing cost
- Cost of potential disaster: Cost of recovered disasters and also cost of unrecoverable disasters.

*4.3 Failure Detection*

Failure detection time strongly affects on the system downtime, so it is critical to detect and report a failure as soon as possible for a fast and correct DR. On the other hand, in multiple backup sites there is a major question: How to distinguish between network failure and service disruption.

*4.4 Security*

As mentioned before, DR can be created by nature or can be human-made. Cyber-terrorism attack is one of human-made disasters which can be accomplished for many reasons. In this case, protection and recovery of important data will be a main goal in DR plans beside of system restoration.

*4.5 Replication Latency*

DR mechanisms rely on replication technique to make backups. Current replication techniques are classified into two categories: synchronous and asynchronous (Ji et al., 2003). However, both of them have some benefits and some flaws. Synchronized replication, guarantees very good RPO and RTO, but it is expensive and also can affect on system performance because of large overhead. This issue is more serious in multi-tier web applications, because it can significantly increase Round Trip Time (RRR) between primary and backup site. On the other hand, a backup model adopted with async replication is cheaper and also system suffers low overhead, but the quality of DR service will be decreased. Therefore, trading off between cost, performance of the system and also replication latency is an undeniable challenge in cloud disaster solutions.

*4.6 Data Storage*

Business database storage is one of the problems of enterprises which can be solved by cloud services. By increasing of cloud usage in business and market, enterprises need to storage huge amount of data on cloud-based storages. Instead of conventional data storage devices, cloud storage service can save money and is also more flexible. The architecture of a cloud storage system includes four layers: physical storage, infrastructure management, application interface and access layer. In order to satisfy applications and also to guarantee the security of data, computing has to be distributed but storage has to be centralized. Therefore, storage single point of failure and data loss are critical challenges to store data in cloud service providers (Pokharel et al., 2010).

### *4.7 Lack of Redundancy*

When a disaster happens, primary site becomes unavailable and secondary site has to be activated. In this case, there is no ability to sync or async replication in a backup site but data and system states only can be stored locally. It is a serious threat to the system. This issue is temporary and will be removed after recovery of the primary site. However, to achieve the best DR solutions, especially in high availability services (such as business data storage), it is better to consider all risky situations.

## **5. DR Solutions**

In this section, we will discuss some DR solutions which have been proposed to overcome the problems and challenges in cloud-based DR.

### *5.1 Local Backup*

A solution for dependency problem has been proposed in (Javaraiah, 2011). A Linux box can be deployed on the side of customers to make control of data and to get backup of both data or even complete application. Local storage can be updated through a secured channel. By this technique, migration between cloud service providers and also migration between public to private, and private to public is possible. In the event of a disaster, local backup can provide the services that were served by the service provider.

### *5.2 Geographical Redundancy and Backup (GRB)*

Although geographical redundancy can be used in traditional model, but it is expensive and unaffordable. In (Pokharel et al., 2010), two cloud zones have a replication of each other. If one zone becomes down, then another zone will be on and provide the services. There is a module that monitors the zones to detect disaster. Primary zone has an active load balancer to request extra resources or even released unused resources. Second zone also has a passive load balancer. Another research (Khan and Tahboub, 2011) has been proposed a method to select optimal locations for multiple backup. The number of places is decided based on the nature of application and priority of services. Distance and bandwidth are two factors to choose the best sites in this method. However, this work neglects some critical factors such as the capacity of mirror sites and the number of node sources which can be hosted in each location.

### *5.3 Inter-Private Cloud Storage (IPCS)*

This approach was proposed for cloud data storage (Jian-hua and Nan, 2011). According to Storage Networking Industry Association (SNIA), at least three backup locations are necessary for business data storage. Users' data should be stored in three different geographical locations: Servers, Local backup server (LBS) and remote backup server (RBS). The private clouds are established for any enterprises consist some servers and an LBS; and also an inter-private cloud storage is created in a public cloud consists the RBSs to be shared between public clouds. This model gives communication ability to backup locations in order to increase data integration. Figure 3 shows the architecture of this approach.

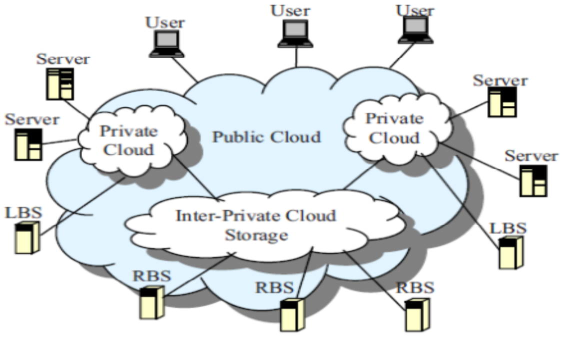

Figure 3. Inter-private cloud architecture (Jian-hua and Nan, 2011)

### *5.4 Resource Management*

Heterogeneous clouds consist many different hardware and software such as hybrid storage and diverse disks. In cloud-based enterprises, entire business data are stored in the cloud storage. So, data protection, safety and recovery are critical in these environments. Data in danger is the data which has been processed at the primary host but has not taken place in the backup host yet. So, in the case of disaster, It is necessary to use enhanced technology for data recovery in storage clouds. There are three solutions for data recovery proposed in (Patil et al., 2012):

- Using fastest disk technology in the event of a disaster for replication of data in danger.
- Changing dirty page threshold: The percentage of dirty pages in RAM which have to be waited for flushing to disk might be reduced (Rudolph, 1990).
- Prediction and replacement of risky devices: Some important factors such as power consumption, heat dissipation, carbon credit utilization and importance of data (stored on each disk) can be calculated in a specific period of time. By these factors, a mathematical equation will be formed to make a replace priority list.

### *5.5 Secure-Distributed Data Backup (SDDB)*

An innovative technique has been presented in (Ueno et al., 2010) to protect data in the event of disaster. The data protection technique has six stages:

- First data encryption: Data has to be encrypted after receiving into a data center.
- Spatial scrambling: By a spatial scrambling algorithm, the order of data files is changed.
- Fragmentation, duplication : Data files are divided into some fragments and these fragments are duplicated in terms of service level agreement.
- Second encryption: Fragments are encrypted again with a different key.
- Shuffling & Distribution: In the last stage, fragments are distributed using a shuffling method into unused memory capacities.
- Transferring Metadata to backup server: Metadata including encryption keys, shuffling, fragmentation and distribute information is sent to a supervisory server.

If a disaster happens, the supervisory server will gather all information from distributed devices and performs decryption (2nd), sort & merge, inverse spatial scrambling and decryption (1st), respectively.

### *5.6 Pipelined Replication*

This replication technique (Wood et al., 2011) aims to gain both the performance of async replication and the consistency of sync replication. In sync replication, processing cannot continue until replication is completely finished at the backup site. Whereas, in async replication, after storing data in the local storage the process can be started. The result can be replied to the client, and then the writes are replicated to the backup site in an epoch. Pipelined replication performs replication and process in parallel as in the following scenario.

- Scenario of usage: The client sends a request to the web server. The web server processes the requests, then sends data to the local database in the primary data center. At this stage, the writes are flushed in the remote backup site, and the process operation can be performed in parallel. However, the reply to the client can be committed only after receiving the Ack from the backup site. Therefore, Pipeline replication facilitates replication procedure, and also guarantees the writes protection.

### *5.7 Scale Up/Down*

Sometimes, performing functions with high priority can decrease money loss or even increase the revenue in the event of a disaster. Priority of services can be defined by some different features such as service level agreement, and the amount of revenue and urgent needs. After a natural disaster occurs in an area, cloud service providers are faced with flooding service requests. In this case, service providers have to manage their existent users' services and also handle new user requests. Service providers must satisfy existent users and should serve to new customers as much as possible. In (Nakajima et al., 2013), a management engine has been introduced for carrier networks. In case of a large scale natural disaster (like earthquakes), this system uses a DR scenario by scaling up resources for the high-priority services (e.g., voice communication) and scaling down allocated resources to low-priority service (e.g., video on-demand).

### *5.8 Dual-Role Operation*

For increasing utilization of recourses, (Aghdaie and Tamir, 2003) introduces a simple technique. As shown in Figure 4, in this technique each host can operate as the primary host for some applications and can also be the backup host for some other applications. In this architecture, clients send their requests to the backup host first, then the backup host transmits those requests to primary host. After processing, primary host sends a log to the backup and finally reply to the clients. When a failure happens, the primary host becomes unavailable, and backup host has to handle the requests of the failed host. However, this technique cannot guarantee a good service restoration by itself, because backup site must share the resources between its own requests and redirected requests.

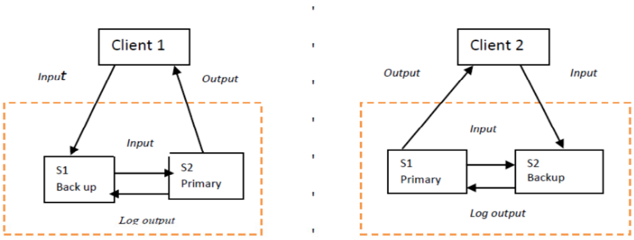

Figure 4. Dual role operation failover (Aghdaie and Tamir, 2003)

Table 6 shows an overview of common challenges and related solutions in DR plans which have been discussed in two last sections.

Table 6. Main challenges and related solutions

| Challenges | Solutions | Technique |
|---|---|---|
| **Dependency** | Local backup | • Using a Linux box at the customer premises |
| **Cost** | Scale up/down,<br>Dual-Role operation | • Allocating resources to high priority services.<br>• Hiring and running idle physical nodes on the secondary site |
| **Failure prediction and detection** | Resource Management ,<br>GRB | • Prediction and replacement of risky hardware<br>• Using monitoring unit |
| **Security** | SDDB | • Using encryption, scrambling and shuffling techniques |
| **Replication latency** | Pipelined replication | • Performing replication and process operations in parallel |
| **Data storage** | IPCS | • Using an inter private cloud |
| **Lack of redundancy** | GRB, IPCS | • Multiple backup |

## 6. Disaster Recovery Platforms

In this section, different cloud-based DR systems will be introduced briefly. Also benefits and weaknesses of each system will be discussed.

### *6.1 SecondSite*

The SecondSite (Rajagopalan et al., 2012) is a disaster tolerance as a service system cloud. This platform is intended to cope three challenges: 1. Reducing RPO 2. Failure detection 3. Service restoration. For this reason, it uses three techniques:

- Using a storage to keep writes between 2 checkpoints: Checkpoints move between sites in a specific period. However, if a failure happens in this time, some data will be lost. For this reason, a Distributed Replicated Block Device (DRBD) (Reisner and Ellenberg, 2005) is used to store replications in both synchronous and asynchronous modes.
- Using a quorum node to detect and distinguish a real failure: A quorum node has been designed to monitor primary and backup server. If replications have not been received by the backup site in the waiting time, backup site sends a message to quorum node. In this case, if the quorum node receives a heartbeat form primary node, it means primary server is active and the replication link has a problem; otherwise the backup site will be active.
- Using a backup site: There is a geographically separated backup site which allows to replicate groups of virtual machines through wide-area Internet links. SecondSite increases ability to fast failure detection and also differentiate between network failures and host failures. Using DRDB, resynchronize storage can be

done for recovering primary site without VMs interruption in the backup site.

Although, SecondSite is not suitable for stateless services, however it increases availability for small and medium businesses.

### *6.2 Remus*

Remus (Cully et al., 2005) - based on Xen hypervisor (Barham et al., 2003) - is a high availability cloud service to tolerate disaster using storage replication combined with live VMs migration. In this system, a protected software is encapsulated in the virtual machines to asynchronously replicate whole-system checkpoints in a backup site with a high frequency. It is assumed that both replicas are in the same local area network (LAN). Remus pursues three main goals: 1. Providing low-level service to gain generality 2. Transparency 3. Seamless failure recovery.

Remus uses an active primary host and a passive backup host to replicate checkpoints. All writes have to be stored in backup RAM until a checkpoint completes. Migrated Virtual machines execute on the backup only if a failure is detected. Remus consists of 4 stages:

- Stop running VMs and propagate only changed states into a buffer
- Transmission of buffered states into backup RAM
- Send an ACK message to primary host after checkpoint completion
- Release the network buffer to external clients.

This system integrates a simple failure detection into the checkpoint process. If checkpoints are not received by the backup site in an epoch, backup site will be active, on the other hand, if backup response is not received during a specific period, then primary site will suppose a failure at the backup host. However, Remus increases performance overhead which leads to some latency, because it requires to ensure consistent replication. In addition, this system needs a significant bandwidth.

### *6.3 Romulus*

Romulus (Caraman et al., 2009) has been designed as a disaster tolerant system based on the KVM hypervisor (Kivity et al., 2007). This platform is an extension of Remus system. Romulus provides an accurate algorithm for disaster tolerant in seven stages in details, which are:

- Disk replication and network protection
- VM checkpoint
- Checkpoint synchronization
- Additional disk replication and network protection
- VM replication
- Replication synchronization
- Failure detection and failover.

The flaw of Remus is that it uses one buffer to replicate writes between primary host and backup. Happening a failure in this buffer before transferring checkpoint causes an inconsistency between disk and VM state; and it can break fault tolerance of Remus. For this reason, Romulus uses a new buffer to replicate disk writes after any checkpoint. Second flaw is that network egress traffic cannot be released until completely transferring checkpoint to storage backup host which can decrease system performance. However, Romulus uses a new egress traffic buffer to solve this problem. Romulus can tolerate failure in two situations:

On the fly: it consists disk and VM state replication into a new writes buffer during VM running.

Failover: the ability of service recovery after a disaster.

### *6.4 DT Enabled Cloud Architecture*

It is an extended architecture based on Romulus seven stage algorithm. It uses a hierarchical tree architecture (Caraman et al., 2012) based on the Eucalyptus IaaS architecture (Daniel et al., 2009). It provides a disaster tolerant service with respect to resource allocation issue which is a challenge in DT services. Host and backup clusters are monitored by high availability controllers. Each cluster has three different controllers:

- Storage controller: To control and manage the cluster storage.
- Cluster controller: To manage IPs, centralized memory and CPU availability.

- Node controller: To load, start and stop the VMs.

Different nodes and also different clusters can communicate with each other for better resource allocation. For this purpose, backup cluster controller allocates a VM to a node. Then, node controller loads and starts the VM and allocates it to the primary host. Finally, primary node controller loads and starts the VM.

In this system, VM failover consists of two scenarios. The first scenario is cluster failure. In this situation, backup cluster will be activated. Node failure is another scenario in which cluster controller releases VMs' IP and allocates a backup node to compose required VMs. This system is most useful for extended distance and metropolitan clusters because of low latency requirements.

### *6.5 Kemari*

Kemari (Tamura et al., 2008) is a cluster system which tries to keep VMs transparently running in the event of hardware failures. Kemari uses primary-backup approach so that any storage or network event that changes the state of the primary VM must be synchronized in backup VM. This system has gained the benefits of the Lock stepping (Bressoud and Schneider, 1996) and the Checkpointing - two main approaches for synchronizing VM state- which are:

- Less complexity compared to lock stepping approach.
- It does not need any external buffering mechanisms which can affect on output latency.

### *6.6 RUBiS*

RUBiS (Wood et al., 2010) is a cloud architecture aims to both DR and also minimizing costs with respect to Service Level Agreement. As shown in Figure 5, in ordinary operation, a primary data center including some servers and a database accomplish normal traffics. A cloud is in charge of disaster recovery with two types of resources: Replication mode resources for getting backup before a disaster which is active; and failover mode resources that will be activated only after a disaster. It is notable that service providers can rent the inactive resources to other customers for revenue maximization. In the case of a disaster, leased resources must be released and allocated to the failover procedure.

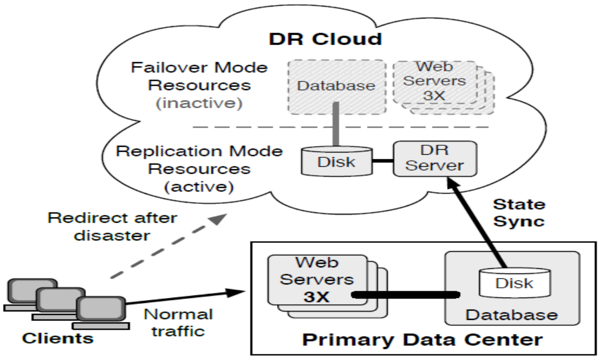

Figure 5. Overviews of RUBiS system architecture (Wood et al., 2010)

### *6.7 Taiji*

Taiji (Zhu et al., 2011) is a Hypervisor-Based Fault Tolerant (HBFT) prototype which uses a mechanism similar to Remus. However, instead of Remus which uses separated local disk for replication, Taiji uses a Network Attach Storage (NAS). Shared storage may become a single point and cause a weakness of this method, so RAID (Patterson et al., 1988) or commercial NAS (Synology, online) solution should be deployed. On the other hand, because of using shared storage, the need of synchronizing is decreased and also file system state is maintained in the event of disaster.

### *6.8 HS-DRT System*

The goal of the HS-DRT system is protecting important data from natural or subversive disasters. This system (Ueno, 2010) uses a HS-DRT processor -which we described as SDDB (section 6, part 5) - with a cloud computing system. Clients are as terminals which request some web applications. The HS-DRT processor has functioned as a web application and also encryption, spatial scrambling, fragmentation of data. At the end, data is sent and stored in a private or public cloud. The system architecture is shown in Figure 6. This system severely increases security of data before and after disaster in cloud environments. However, It has two weaknesses:

- The performance of the web application will be decreased if the number of duplicated copies increases.
- This system cannot guarantee consistency between different copies of file data.

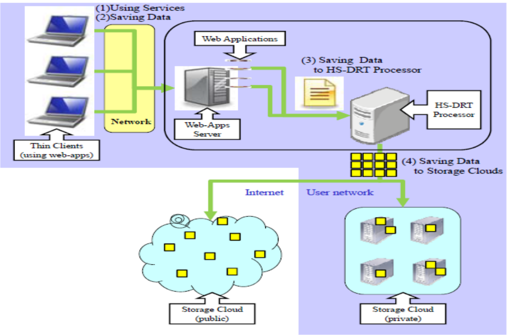

Figure 6. The architecture of the HS-DRT system (Ueno et al., 2010)

### *6.9 PipeCloud*

This cloud-based multi-tier application system uses the Pipelined replication technique (as mentioned in the last section) as a DR solution. PipeCloud architecture is composed of a cloud backup site and a primary data center (Wood et al., 2011). The goal of this system is mirroring storage to the backup site and minimizing RPO. The main tasks of PipeCloud are:

- Replicating all disk writes to a backup site by the replication technique
- Tracking the order and dependencies of the disk writes
- Releasing network packets only after storing the disk writes on the backup site.

This system results in a higher throughput and lower response time by decreasing the impact of WAN latency on the performance. For this purpose, the system overlaps replication with application processing. Also, it guarantees zero data loss consistency. However instead of Remus, PipeCloud cannot protect the memory states because it leads to large overhead on WAN.

### *6.10 Disaster-CDM*

Huge amount of disaster-related data have been generated by government, organization, automation systems and even social media (Grolinger et al., 2013). aims to provide a Knowledge as a Service KaaS) framework for disaster cloud data management which can lead to better preparation, response and recovery of disasters.

As shown in Figure 7, this system uses both cloud storage and NoSQL (Schram and Anderson, 2012) to store data. Disaster-CDM consists two parts:

- Knowledge acquisition: Obtaining knowledge from a variety of sources, processing and storing in datacenters.
- Knowledge delivery service: Merging information from diverse databases and delivering knowledge to users.

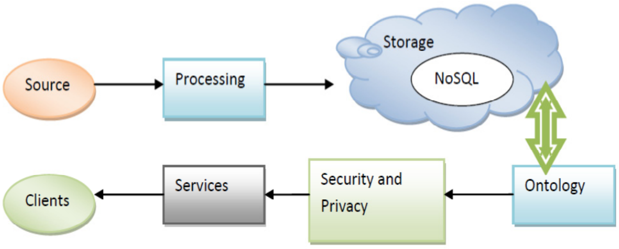

Figure 7. Disaster-CDM Framework

### *6.11 Distributed Cloud System Architecture*

In (Silva et al., 2013) the authors have introduced a cloud system to provide high dependability of the system based on severe redundancy. The system has multiple datacenters which are geographically separated from each other. Each datacenter includes both hot and warm physical nodes. VMs are active in both warm and hot physical nodes but only running in the hot nodes. In order for DR, there is a backup server which stores a copy of each VM.

When a physical node failure occurs, the VMs migrate to a warm physical node. In the case of a disaster which makes a data center unavailable, backup site transmits VM copies to another data center. Although this system architecture is expensive, but it highly increases the dependability which can be adequate for Infrastructure as a Service (IaaS) clouds. In addition, this paper has introduced a hierarchical approach to model cloud systems based on dependability metrics as well as disaster occurrence using the Statistic Petri Net approach (German, 2000). Figure 8 shows the architecture of this DR system.

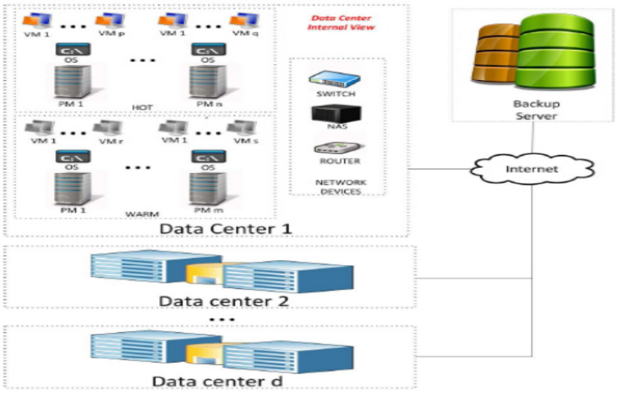

Figure 8. Distributed cloud system architecture (Silva et al., 2013)

Table 7 shows an overall comparison of different cloud-based DR platforms in terms of 10 key properties.

Table 7. Comparing cloud-based DR platforms in terms of different properties

| Cloud-based DR systems | User premises backup | Dual-role operation | Multi-tier | Multiple backup | Shared data storage | Security techniques | Quorum host | Live VM migration | Knowledge-based DR service | Pipeline replication |
|---|---|---|---|---|---|---|---|---|---|---|
| SecondSite | ✓ | ✓ | | ✓ | | | ✓ | | | |
| Remus | | | | | | | | ✓ | | |
| Romulus | | | | | | | | ✓ | | |
| DT enabled cloud architecture | | | | | | | | ✓ | | ✓ |
| Kemari | | | | | | | | ✓ | | |
| RUBiS | | | ✓ | | | | | | | |
| Taiji | | | | | ✓ | | | ✓ | | |
| HS-DRT system | | | ✓ | ✓ | | ✓ | | | | |
| PipeCloud | | | | | | | | ✓ | | ✓ |
| Disaster-CDM | | | | | | | | | ✓ | |
| Distributed cloud system architecture | | ✓ | | | ✓ | | | | | |

## 7. Open Issues and Future Directions

In the last sections, we described the main properties and challenges of DR systems. Then, some related solutions and systems have been introduced. However, some issues still require more effort to reach a worthy level of DR mechanisms in cloud computing. In this section, we introduce some open and related issues in this area:

### *7.1 Maximizing Resource Utilization*

Cloud customers pay for DR resources only after a disaster happens. However, these resources must be always available when needed. Since disasters are usually scarce, the revenue of the DR servers is less. Therefore, CSPs need the ways to both increase the utilization and revenue of DR servers and also guarantee DR services, simultaneously.

### *7.2 Correlated Failures*

Occurring disaster in a specific area can lead to vast service interruption, and consequently, many customers have to be recovered by CSPs. In this case, it is possible that related servers cannot be able to handle all the customers. So, it can be critical to multiplex customers of the same area in different servers. One major challenge in this case is how to distribute customers between cloud servers to minimize correlated failure risk with respect to required QoS for each server and also cloud SLA (Wood et al., 2010).

### *7.3 Privacy and Confidentiality*

In the event of disaster, the private data centers of enterprises would be failover by cloud environments. So, one critical issue is that cloud must guarantee confidentiality of data and privacy of resources which are used for DR. On the other hand, the cloud has to guarantee the performance of applications would not be affected by other disasters happened to other enterprises.

### *7.4 Failover and Failback Procedure*

Failover and failback procedure are two important stages in DR mechanism. Failover procedure is performed to automatically switch over to a backup site whenever the current active site becomes unavailable. In the event of a disaster, failover procedure excludes failed resources and redirect workloads to a secondary site using a specialized load balancer. Client-transparent procedure and fast IP failover requirements are two main challenges in this issue. On the other hand, after passing disaster, application control has to be reverted to the original site. For this purpose, bidirectional state replication must be supported by the DR mechanism. A portion of data may be lost because of the disaster in the primary site and also new data will be created in the backup site. Therefore, one major challenge is that how to determine new and old data which must be resynchronized to the primary site (Aghdaie and Tamir, 2003).

### *7.5 Disaster Monitoring*

Since failure tolerance is necessary to deliver expected QoS, it will be essential to determine which processes are operational and which crashed in the cloud systems. In the case of disaster, the sooner failure detection in either primary site or backup site leads to better RTO. So, the challenge is how should the status of cloud be monitored and how a disaster can be detected in its early stages (Aceto et al., 2013).

### *7.6 Resource Scheduling*

The number of cloud-based services are increasing day by day and so has increased the complexity of cloud infrastructures. Hence, resource scheduling is a critical issue in the modern cloud environments. This issue is more crucial for cloud-base DR platforms since they face unpredictable arrival rate and have to consider a variety of catastrophic situations. Building on this, more efficient resource scheduling techniques are needed in order for current DR platforms to be also optimal.

## 8. Conclusion

In this paper, we have provided an in depth analysis of the state of the art for DR in cloud computing. First, we briefly introduced cloud computing, including background, properties, advantages and challenges. Then, we discussed the details of cloud-based disaster recovery and compared it with traditional approaches. In addition, we also derived the main challenges of DR mechanisms and proposed solutions to overcome them. Furthermore, the main DR platforms are discussed, followed by open issues and future direction in the field of cloud-based DR mechanisms. Finally, a DR procedure is proposed which can effectively utilized by any DR mechanism.

## References

Aceto, G., Botta, A., Donato, W., & Pescape, A. (2013). Cloud monitoring: A survey. *Computer Networks*, *57(9)*, 2093-2915. http://dx.doi.org/10.1016/j.comnet.2013.04.001

Aghdaie, N., & Tamir, Y. (2003). Fast transparent failover for reliable web service. *15th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)* (pp. 757-762).

Alhazmi, O. H., & Malaiya, Y. K. (2012). Assessing Disaster Recovery Alternatives: On-site, Colocation or Cloud. *IEEE 23rd International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 19-20). http://dx.doi.org/10.1109/ISSREW.2012.20

Alhazmi, O. H., & Malaiya, Y. K. (2013). Evaluating disaster recovery plans using the cloud. Reliability and Maintainability Symposium (RAMS), *IEEE Proceedings-Annual* (pp. 1-6). http://dx.doi.org/10.1109/RAMS.2013.6517700

Arean, O. (2013). Disaster recovery in the cloud. *Network Security, 9*, 5-7. http://dx.doi.org/10.1016/S1353-4858(13)70101-6

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... Zaharia, M. (2010). Above the Clouds : A View of Cloud Computing. *Communications of the ACM*, *53(4)*, 50-58. http://dx.doi.org/10.1145/1721654.1721672

Azagury, A., Factor, M. E., & Satran, J. (2002). Point-in-time copy: Yesterday, today and tomorrow, *Proceedings of the IEEE/NASA MSST* (pp. 259-270).

Barham, P., Dargovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., ... Warfield, A. (2003). Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review, 37*(5), 164-177. http://dx.doi.org/10.1145/945445.945462

Bressoud, T. C., & Schneider, F. B. (1996). Hypervisor-based fault tolerance. *ACM Transactions on Computer Systems*, 14(1), 80–107.

Buyya, R., Broberg, J., & Goscinski, A. M. (2011), Cloud computing: Principles and paradigms, Wiley.

Caraman, M. C., Moraru, S. A., Dan, S. & Kristaly, D. M. (2009). Romulus, Disaster Tolerant System based on Kernel Virtual Machines. *20th International DAAAM Symposium :Intelligent Manufacturing & Automation: Theory, Practice & Education* (pp. 1671-78).

Caraman, M. C., Moraru, S. A., Dan, S., & Grama, C. (2012). Continuous Disaster Tolerance in the IaaS clouds. *13th IEEE International Conference on Optimization of Electrical and Electronic Equipment (OPTIM),* (pp.1226-32). http://dx.doi.org/10.1109/OPTIM.2012.6231987

Cloud taxonomy. Retrieved from http://cloudtaxonomy.opencrowd.com.

Cully, B., Lefebvre, G., Meyer, D., Fraser, K., Hutchinson, N., & Warfield, A. (2008). Remus: High Availability via Asynchronous Virtual Machine Replication. *5th USENIX Symposium on Networked Systems Design and Implementation* (pp. 161-174).

Daniel, N., Rich, W., Chris, G., Graziano, O., Sunil, S., Lamia, Y., & Dmitrii, Z. (2009). The Eucalyptus Open-source Cloud-computing System. *9th IEEE International Symposium on Cluster Computing and the Grid* (pp.124-131). http://dx.doi.org/10.1109/CCGRID.2009.93

German, R. (2000). Performance Analysis of Communication Systems with Non-Markovian Stochastic Petri Nets, Wiley & Sons.

Grolinger, K., Capretz, M. A. M., Mezghani, E., & Exposito, E. (2013). Knowledge as a Service Framework for Disaster Data Management. *22nd IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE),* (pp.313-318). http://dx.doi.org/10.1109/WETICE.2013.48

Guster, D., & Lee, O. F. (2011). Enhancing the Disaster Recovery Plan Through Virtualization. *Journal of Information Technology Research, 4*(4), 18-40. http://dx.doi.org/10.4018/jitr.2011100102

IBM white paper. (2012). Virtualizing disaster recovery using cloud computing, *IBM global technology services*.

Jaiswal, V., Sen, A., & Verma, A. (2011). RSCMap: Resiliency planning in storage clouds. *ICSOC'11 Proceedings of the 9th international conference on Service-Oriented Computing* (pp. 505-512).

Javaraiah, V. (2011). Backup for cloud and disaster recovery for consumers and SMBs. *IEEE 5th International Conference on Advanced Networks and Telecommunication Systems (ANTS)* (pp. 1-3). http://dx.doi.org/10.1109/ANTS.2011.6163671

Ji, M., Veitch, A., & Wilkes, J. (2003). Seneca: Remote Mirroring Done Write. Proceedings *of USENIX Technical Conference* (pp. 253-268).

Jian-hua, Z., & Nan, Z. (2011). Cloud Computing-based Data Storage and Disaster Recovery. *IEEE International Conference on Future Computer Science and Education (ICFCSE),* (pp. 629-632). http://dx.doi.org/10.1109/ICFCSE.2011.157

Kashiwazaki, H. (2012). Practical uses of cloud computing services in a Japanese university of the arts against aftermath of the 2011 Tohoku earthquake. *Proceedings of the ACM SIGUCCS 40th annual conference on Special interest group on university and college computing services* (pp. 49-52).

Khan, J. I., & Tahboub, O. Y. (2011). Peer-to-Peer Enterprise Data Backup over a Ren Cloud. *IEEE 8th International Conference on Information Technology: New Generations (ITNG)* (pp. 959-964). http://dx.doi.org/10.1109/ITNG.2011.164

Kivity, A., Kamay, Y., Laor, D., Lublin, U., & Liguori, A. (2007). kvm: the Linux Virtual Machine Monitor. *Proceedings of the Linux Symposium* (pp. 225-230).

Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). NIST Cloud Computing Reference Architecture. *NIST Special Publication* (pp.500-292).

Lwin, T. T., & Thein, T. (2009). High Availability Cluster System for Local Disaster Recovery with Markov Modeling Approach. *International Journal of Computer Science Issues, 6*(2), 25-32. http://arxiv.org>CS>arx17:0912.1835

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing, NIST Special Publication, Special publication 800-145. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

Nakajima, Y., Masutani, H., Shen, W., & Tanaka, H. (2013). design and implementation of virtualized ict resource management system for carrier network services toward cloud computing era. *ITU Kaleidoscope: Building Sustainable Communities (K-2013*) (pp.1-8).

Nayak, T., Routray, R., Singh, A., Uttamchandani, S., & Verma, A. (2010). End-to-end Disaster Recovery Planning: From Art to Science. *IEEE Network Operations and Management Symposium* (pp. 357-364). http://dx.doi.org/10.1109/NOMS.2010.5488491

Patil, S. R., Shiraguppi, R. M., Jain, B. P., & Eda, S. (2012). Methodology for Usage of Emerging Disk to Ameliorate Hybrid Storage Clouds. *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)* (pp.1-5). http://dx.doi.org/10.1109/CCEM.2012.6354615

Patterson, D. A., Gibson, G., & Katz, R. H. (1988). A Case for Redundant Arrays of Inexpensive Disks (RAID). *International Conference of Management of Data (SIGMOD)* (pp.109-116).

Pokharel, M., Lee, S., & Park, J. S. (2010). Disaster Recovery for System Architecture using Cloud Computing. *10th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT)* (pp. 304-307).

Prakash, S., Mody, S., Wahab, A., Swaminathan S., & Ramani (2012). Disaster recovery services in the cloud for SMEs. *IEEE International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM)* (pp. 139-144). http://dx.doi.org/10.1109/ICCCTAM.2012.6488087

Rajagopalan, S., Cully, B., Connor, R. O., & Warfield, A. (2012). SecondSite: disaster tolerance as a service. *ACM SIGPLAN Notices, 47*(7), 97-107. http://dx.doi.org/10.1145/2365864.2151039.

Reid, S., Kicker, H., Matzke, P., Bartels, A., & Lisserman, M. (2011). Sizing the cloud. Technical report. Retrieved from http://www.forrester.com-/E-/Sizing+The+Cloud/fulltext/RES58161objectid=RES58161

Reisner, P., & Ellenberg, L. (2005). Replicated storage with shared disk semantics. *12th International Linux System Technology Conference* (pp. 1-11).

Rudolph, C. G. (1990). Business Continuation Planning/Disaster Recovery: a Marketing Perspective. *IEEE Communications Magazine, 28*(6), 25-28.

Salapura, V. (2012). Cloud computing: Virtualization and resiliency for data center computing. *IEEE 30th International Conference on Computer Design (ICCD)* (pp. 1-2). http://dx.doi.org/10.1109/ICCD.2012.6378606

Schram, A., & Anderson, K.M. (2012). MySQL to NoSQL: Data Modeling Challenges in Supporting Scalability. *3rd Conference on Systems, Programming, and Applications: Software for Humanity* (pp.191- 202).

Silva, B., Maciel, P., Tavares, E., & Zimmermann, A. (2013). Dependability models for designing disaster tolerant cloud computing systems. *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (pp.1-6). http://dx.doi.org/10.1109/DSN.2013.6575323

Synology. Retrieved from http://www.synology.com

Tamura, Y., Sato, K., Kihara, S., & Moriai, S. (2008). Kemari: Virtual machine synchronization for fault tolerance. *USENIX Annual Technical Conference* (pp.1-2).

Ueno, Y., Miyaho, N., Suzuki, S., & Ichihara, K. (2010). Performance Evaluation of a Disaster Recovery System and Practical Network System Applications. *IEEE Fifth International Conference on Systems and Networks Communications (ICSNC)* (pp.195-200). http://dx.doi.org/10.1109/ICSNC.2010.37

White paper. (2013). *UK Cloud Adaption and Trends*. Retrieved from http://cloudindustryforum.org/white-papers/uk-cloud-adoption-and-trends-for-2013

Wood, T., Cecchet, E., & Ramakrishnan, K. K. (2010). Disaster recovery as a cloud service: Economic benefits & deployment challenges. *2nd USENIX Workshop on Hot Topics in Cloud Computing* (pp. 1-7).

Wood, T., Lagar-Cavilla, H. A., Ramakrishnan, K. K., Shenoy, P., & Van der Merwe, J. (2011). PipeCloud: using causality to overcome speed-of-light delays in cloud-based disaster recovery. *2nd ACM Symposium on Cloud Computing.* http://dx.doi.org/10.1145/2038916.2038933

Zhang, Q., Cheng, L., & Boutaba, R. (2013). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications, 1*(1), 7–18. http://dx.doi.org/10.1007/s13174-010-0007-6

Zhu, J., Jiang, Z., Xiao, V., & Lee, X. (2011). Optimizing the performance of virtual machine synchronization for fault tolerance. *IEEE Transactions on Computers, 60*(12), 1718-1729. http://dx.doi.org/10.1109/TC.2010.224

## Copyrights