IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

REVIEW ON DATA SECURITY AND FAULT TOLERANCE IN CLOUD STORAGE SYSTEM

Kandula Chaithanya Deepthi, Computer Science and Engineering, JNTUHUCEH, Hyderabad

ABSTRACT

Cloud computing is a method for delivering a networked pool of configurable computing resources (such as servers, storage, networks, and services) that can be rapidly provided and transferred with low administration effort. Computing power, storage, platforms, and services are virtualized, dynamically scalable, managed, and given on-demand to cloud customers through the Internet. This is one sort of large-scale distributed computing paradigm. The cloud presents to people as a central hub from which they may access all their anticipated digital resources. With cloud computing, users are able to store data remotely and gain access to on-demand cloud applications without worrying about setting up any specialised hardware or software on their end. Concerns about cloud computing's security and dependability are widespread. The new technology was implemented to address the problems with security and data centre dependability. As a result of the benefits that cloud computing may provide to businesses, more and more of them are beginning to think about adopting cloud strategies and applying them in their operations. Because of the inherent risk of failure in cloud data centres and the critical importance of having constant access to cloud resources, it is imperative that various fault-tolerance strategies be tested and implemented. However, the ever-increasing number of people using cloud storage prompted us to look into fault-tolerance methods, as well as their advantages and disadvantages. After defining fault-tolerance and its application to cloud computing, this paper describes various fault-tolerant approaches and introduces related measures.

Key words: Data Security, Cloud storage, fault tolerance.

1. INTRODUCTION

The term "cloud computing" is used to describe the sharing of server space and other hardware over the Internet. By utilising a service over the Internet at a different location, we can avoid the hassle of storing data on our own hard drive and can avoid changing our software to meet our own demands. Cloud computing is predicated on the principle that existing IT resources should be utilised multiple times. Cloud computing is based on several technologies, including virtualization, distributed computing, utility computing, and, more recently, networking, online, and software services. Third-party remote management of hardware and software is used by both individuals and businesses. Some examples of cloud services include online data storage, social media, electronic mail, and commercial software. Users can take advantage of these services without needing technical knowledge of the underlying infrastructure.

1.1 Cloud Components

There are many parts to cloud computing. Clients, Distributed Servers, and Data Centers are the three main categories that can be used to

categorise the various components, each of which serves a unique function.

• Clients: Typically, these are the computers that are utilised by the end users; in other words, these are the devices that can be used by the end user to handle the information that is stored in the cloud (laptops, mobile phones, PADs etc.)

• Data center: These are the servers that host the service. Using virtualization, a single physical server in a data centre can host several virtual servers.

• Distributed servers: These servers can be found in a variety of locations throughout the world. To put it simply, it makes things easier and safer for the user.

1.2 Significant Objective of study

The purpose of this thesis is to save the datacenter from disaster while also storing the data in a competent manner in cloud data centres using erasure codes such as Reed Solomon codes equipped with data ciphering systems. This will satisfy the fundamental objectives of cloud data centres and achieve the goal of this thesis. The goals are as follows:

- Maintaining storage costs to a minimum.
- Putting together the data centre with as little work as possible.
- Increasing the safety (Access control).
- Getting faster upload and download times.
- Bringing down normalization cost.

2. LITERATURE REVIEW

Cloud computing provides on-demand access to a variety of computing resources in the form of services. It paves the way for companies and individuals to use software without having to physically install it on their own computers, and it provides easy access to any necessary data or programmes via the web. This shift from IT as a product to IT as a service is enabled by the characteristics shown in Fig. 1, including high performance, pay-as-you-go, connectivity, interactivity, reliability, simple programmability, efficiency, scalability, management of large amounts of data, and elasticity (Bokhari et al., 2016).

Fig. 1. Cloud computing.

Cloud computing is a rapidly developing technology that is being adopted by many businesses and corporations to host their critical software. However, there are problems with service reliability and availability for both service providers and consumers because of the widespread adoption of cloud-based services for hosting corporate and enterprise applications (Gokhroo et al., 2017, Nazari Cheraghlou et al., 2016). These problems are inherent to cloud computing due to its decentralised architecture, resource heterogeneity, and large size. As a result, the cloud environment is susceptible to a wide variety of errors that can cause failures and performance degradation.

Following is a list of the most common kinds of faults (Amin et al., 2015; Essa, 2016):

Network fault: As the resources in the cloud are accessed over a network (the Internet), faults with the underlying

network infrastructure are a common source of disruption. Problems like this can arise from a variety of causes, including network partitions, packet loss or corruption, congestion, a failed destination node or link, and so on.

Physical faults: Hardware resource faults include faults with the computer's central processing unit (CPU), memory, storage, power supply, etc.

Process faults: Processes can go wrong for a variety of reasons, including a lack of resources, defects in the software, insufficient processing capacity, and so on.

Service expiry fault: When the service time of a leased resource runs out while the corresponding application is still making use of it, service failures occur.

In the event of a failure, the system will either break down entirely or be forced to shut down. Partially successful attempts are, however, a hallmark of distributed computing and, by extension, cloud computing. When something goes wrong, it could be any part of the network or any part of the process. As a result, rather than a total collapse, we get a partial failure and a decline in performance. While this does produce durable and sturdy systems, suitable fault tolerance methods for HPC are still required. With fault tolerance, the system can still fulfil the user's request even if some of its components are damaged (Charity and Hua, 2016).

3. FAULT TOLERANCE TECHNIQUES

3.1. Fault Classification and the Need for Fault Tolerance in Cloud Computing

In response to increasing service demands, enterprises have built massive data centres. When it came to building data centres, speed and efficiency were once the most important factors to consider. Due in part to the massive amounts of data kept on hand, data centre failures have become increasingly common in recent years, what with the rise of cloud computing and the popularity of cloud services. The more data there are, the more hard it is to gain access to them, and separate permissions may be needed for different applications or data items. Every system should be as resilient and trustworthy as possible, and that's what fault tolerance is all about. Two broad classes of approaches exist, "proactive" and "reactive," that are defined by the fault tolerance policies and procedures they employ.

In order to avoid problems in the future, I employ a policy of proactive fault tolerance, which involves using prediction to avoid retrieval of fault, error, and failure and then detecting the suspicious items and replacing them with the proper data. The goal of a reactive fault-tolerance strategy is to mitigate errors as they occur. There are two main categories: fault-treatment methods and error-processing algorithms. Error processing aims to rectify computation mistakes. The goal of error therapy is to stop errors from being activated again.

3.2. Existing Fault Tolerance Techniques in Cloud Computing

There have been a lot of studies on data fault-tolerant systems in the recent years, leading to the development of novel approaches to assessing the merits and drawbacks of such systems. This section presents the most up-to-date fault-tolerance methods currently available for Cloud Computing.

Check pointing–It is a reliable method of fault tolerance at the task level for large and persistent programmes. Here, a check point is performed after each time the system is modified. When a work fails, it can be restarted from its most recent saved point rather than from scratch.

Job Migration –It has occurred on occasion that a work just cannot be finished on a certain machine. If a process fails, it can be moved to another computer. Incorporate job migration with HA-Proxy.

Replication- It is a major fault-tolerant technology in data centres and sees extensive use in both research and online service environments. It is to copy something that is meant to be replicated. Replication relies on a

variety of resources to carry out its many responsibilities and produce the best possible outcomes. HAProxy, Hadoop, and Amazon EC2 are all viable options for implementing replication.

Self-Healing- It is possible to break down a large job into smaller tasks. Increased efficiency is the result of these multiplications. Application failures are automatically handled when many instances of the same application are operating on separate virtual machines.

3.3 Proactive approaches

Following are a few examples of proposed methods and frameworks for proactive fault tolerance:

An autonomous migration of virtual machines (VMs) from failing physical machines (PMs) in the data centre (DC) to some optimal target PMs was presented by Jialei Liu et al. in 2016 as part of the PCFT (Proactive Co-ordinated fault tolerance) technique. The initial distribution of virtual clusters has also been proposed as the subject of an algorithm.

The two-stage method that is being proposed is as follows: An initial step in predicting a failing hardware system is the proposed CPU temperature-based architecture. In the second stage, a metaheuristic algorithm called the Particle Swarm Optimization method is used to seek for the best possible target physical machine. The authors have used appropriate measures to quantify the efficacy of the suggested method and have compared it to five other models: FF (First-fit), BF (Best-fit), RFF (Random First-Fit), MBFD (Modified Best Fit Decreasing), and IVCA. Experiments showed that because PCFT employs an enhanced version of the PSO (Particle swarm optimization) method, it can reduce transmission overhead and reduce overall execution time compared to the other five algorithms. When compared to the other five methods, the PCFT uses fewer of the network's edge, aggregation, root switch, and overall resources.

The SVM-Grid online FD (fault detection) method was introduced by Zhang et al. (2018). In terms of cloud reliability, this method has been called crucial. The author claims that many fault detection models are needed to learn more about the cloud system's inner workings. Traditional SVM (support vector machine) models are widely employed, despite their low accuracy. A reliance on the SVM-Grid is recommended for an online fault detection model as a solution to this problem. SVM-Grid is able to foresee future cloud-related problems. Using the grid method, we were able to fine-tune the model's prediction for improved accuracy by adjusting one of its input parameters. A fine-tuned prediction algorithm and an updated FT algorithm for example DBs (databases) have also been created to improve fault prediction performance and reduce time cost. Google's own dataset has been used in simulated tests (Google2 application cluster). A number of other methods, including back propagation, LVQ (Learning vector quantization), and classic SVM, have been compared to the suggested method. The experimental results showed that the newly constructed model outperformed BP, LVQ, and classic SVM in terms of accuracy while also being more efficient in terms of time investment.

3.4 Reactive approaches

Below are some examples of reactive fault tolerance-based models and frameworks that have been proposed:

An OPVMP (optimal redundant virtual machine placement) model was proposed by Wang et al. in 2016. This strategy, based on replication, was designed to make server-based cloud applications more reliable. The three components of the proposed method are host server selection, virtual machine (VM) placement, and recovery strategy choice. It has been determined that a heuristic method is the best way to choose host servers and position virtual machines. Results from studies conducted in the CloudSim simulator prove the method's efficacy (Calheiros et al., 2009). The proposed method's findings have been compared to those of five different models. The experimental data

showed that the proposed method consumed less bandwidth than the competing techniques.

An Edge switch failure aware checkpointing (EDCKP) model was suggested by Zhou et al. in 2017. Improved service reliability in the cloud is one of the primary motivations for the development of this paradigm. There has been some thought given to the fat tree network topology, and two techniques have been proposed to deal with the problem of edge switch failure. In one method, a storage server is chosen to hold the backup copy of the checkpoint image; in another, a recovery server is selected. No checkpoint-based approach (NOCKP) and non-checkpoint-based model (NDCKP) have been used as benchmarks against which the suggested model has been evaluated (is a network topology aware distributed delta checkpoint-based technique). Results from a simulation showed that the EDCKP approach required less time to complete and less network resources to achieve the desired effect of increased service reliability.

4. CLOUD DATA STORAGE CHALLENGES & ISSUES

Cloud computing does not grant access to or management of data stored in the cloud. Since the cloud provider has complete access to the data, they might potentially alter, delete, or copy it without any consequences to the user. Control over virtual computers is guaranteed by the cloud. Figure 2 depicts the generic cloud computing approach, which has less security concerns due to centralised data storage.

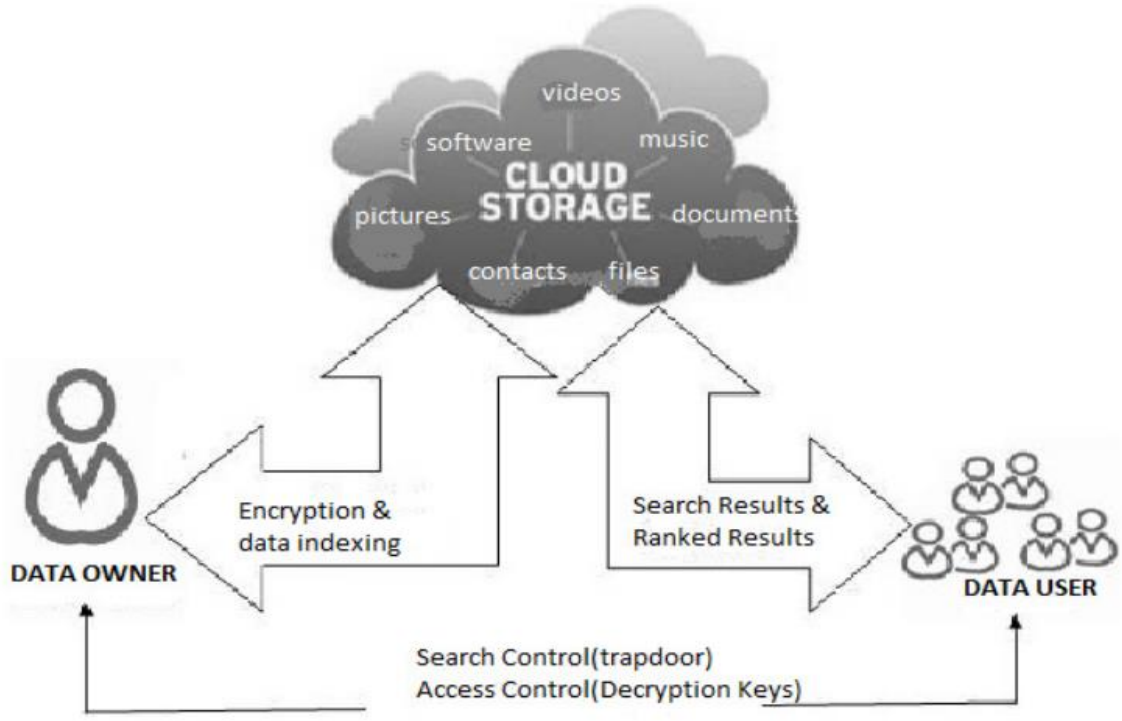

Figure 2: Cloud data storage model.

The only available encryption doesn't provide complete control over the data being saved, although it does provide some improvement over plain text. Cloud computing, with its hallmarks of virtualization and multitenancy, also presents a number of attack vectors beyond those of the conventional cloud architecture. Multiple problems with Figure 3 are outlined and shown below.

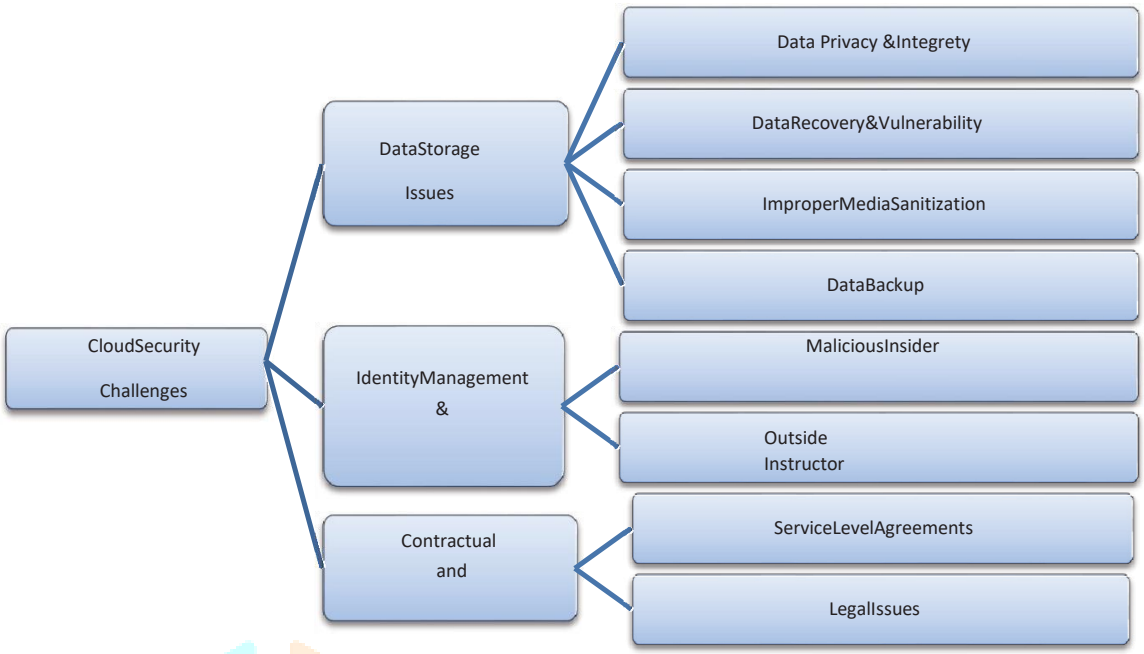

Figure 3. Cloud security Challenges

4.1 Cloud Storage issues

4.1.1 Data privacy and Integrity

The security risks associated with cloud computing outweigh its benefits, which include lower costs and simplified management of shared resources. While it is true that the generic cloud computing paradigm requires protections for data integrity, confidentiality, privacy, and availability, the cloud computing model is more susceptible to security risks. Usership of cloud services and the percentage of applications running in the cloud have both grown rapidly as a result of their relative ease of use. Clients of cloud services have a higher level of risk as a result of these circumstances. If an attack on a data entity is successful, it will result in a data breach, which will allow hackers to access the data of all cloud users. The multi-tenant nature of cloud data was compromised as a result of the integrity breach. In particular, SaaS providers have a high risk of losing their technical data and also face substantial risks associated with storing it. In addition to these dangers, data transformation among many tenants involves a considerable deal of uncertainty. Virtualization allows for the pooling of many different types of physical resources for usage by many different people. As a result, malevolent insiders within the CSP and/or organisation are able to launch attacks. Because of this, a malevolent user might potentially launch attacks on other customers' stored data while processing it. Another significant threat occurs when the CSP sends data to an external storage provider.

4.1.2 Data recoverability and vulnerability

The cloud provides dynamic and on-demand Resource provisioning to the users due to resource pooling and elasticity properties. At a later period, another user may be given the resource that was originally allotted to the first user. Data recovery techniques can be used by a malevolent user to access the information of prior users if memory and storage space are limited.

4.1.3 Improper media refinement.

The reasons for cleaning the storage devices are as follows: (i) the disc may need to be swapped out for another disc (ii) Disk maintenance is unnecessary (or is no longer necessary) (iii) a service massacre Inadequate refinement poses a significant threat to all information. The ability to refine is unavailable in a multi-tenant cloud environment due to the presence of an existing tenant.

4.1.4 Data backup

Data backups are crucial in the event of natural and man-made calamities. To guarantee data availability, the CSP must execute frequent backups. Data backups, in reality, need to adhere to security best practises to stop dangerous actions like tampering and unauthorised access.

5. FAULT–TOLERANCE IN CLOUD COMPUTING

A fault-tolerant system functions normally and reliably despite the presence of defective parts. It's the study and practise of making computer systems that function adequately even when errors occur. Transient, intermittent, or permanent hardware faults, software and design flaws, operator errors, externally caused upsets, or physical damage are only some of the fault types that a fault tolerant system may be able to withstand. The distant processing on computing nodes used by real-time cloud apps increases the risk of inaccuracies. Because of these occurrences, fault tolerance approaches are becoming increasingly important for ensuring the dependability of real-time computing on the cloud.

Described in the following diagram is the interplay between defects, errors, and catastrophic failure. There is a mutual dependence between them. The failure is caused by the occurrence of fault, either in the component (Hardware) or in the designing approaches (Software) that are prone to error.

Figure 4: Generation path of failure

• A system is said to be fail when it will not fulfill the requirements

• An error is the part of the system state that may lead to failure

• The cause of an error is a fault.

5.1 Types of faults

Various criteria allow for the categorization of the defects.

• Network fault: An fault in a network caused by factors such as congestion, lost data, corrupted packets, failed destinations, broken links, etc.

• Physical faults: Hardware failures including faulty central processing units (CPUs), memory, storage, or even a loss of power might cause this error.

• Media faults: Fault caused by a crashed media head in a storage device.

• Process faults: A fault caused by a factor other than intended behaviour, such as a lack of resources, a malfunctioning piece of software, insufficient computing power, etc.

• Service expiry fault: During the time an application is utilising a resource, that resource's service time could end.

It is possible to categorise a fault according to the amount of time and processing power required to fix it. Timing failure, omission failure, response failure, and crash failure are four types of errors that can occur when using system resources to perform computation. It's possible to have persistent, periodic, or temporary problems.

• Permanent failure: Circumstances like power outages, natural disasters, and human error all play a role in these faults. Due to these failures, normal system operation may be severely disrupted.

• Intermittent failure: Occasional occurrences of such failures are to be expected. These breakdowns manifest themselves when the system goes about its business. It is quite

challenging to pinpoint the exact amount of harm produced by these malfunctions.

• Transient failure: There is a simple solution to these problems because they are all the result of a flaw in the system. Retrying the operation or reverting the system to an earlier save point will fix these errors. These kind of problems occur frequently in the computer system.

5.2 Hardware fault-tolerance techniques

- **BIST (Build in Self-test)**

Using this method, the system can analyse any defective propagations by testing them at regular intervals. In the event of a failure, it will automatically replace the faulty part with a spare.

- **TMR (Triple Modular Redundancy)**

In this method, three replicas of a potentially failing part are made and all of them are put through their paces at once. There is a popular vote taken into consideration while choosing them for their work. If there is one problem, it can work around it.

- **Circuit Breaker**

To prevent widespread damage in dispersed systems, this circuit design allows for its interruption.

5.3 Software fault-tolerance techniques

If these methods are used to the software, it becomes more stable.

- **N-Version Programming**

As part of this method, a team of developers creates n distinct iterations of a programme. In this setup, many copies are processed concurrently, and the most robust one is chosen. When the software is still in its developmental stages, this method can be used to detect bugs.

- **Recovery Blocks**

This method is similar to the one described before, with the exception that the redundant copies are not executed simultaneously. Each one is constructed using a unique set of algorithms, and it must be executed individually. This method is implemented when the computation time is shorter than the deadline for a certain task.

- **Check-pointing and Roll-back recovery**

This method performs a test of the system whenever a computation is required.

- **Failure-Oblivious Computing**

This method allows for applications to keep running even if they encounter problems. It deals with invalid memory reads by returning a fabricated value to the application, which then takes into account the new value and disregards the old one in memory. This is in contrast to the previous memory checks, which terminated the programmes when they encountered an invalid input.

- **Recovery Shepherding**

Specifically, the just-in-time binary framework pin is required for this method to function properly. It tethers itself to the faulty application, does an error analysis, keeps account of the fixes made and their outcomes, and finally detaches itself. All of this takes place invisibly in the background while the software continues to run normally and unaffected.

CONCLUSION

The unpredictable nature of the cloud can cause malfunctions and errors in computer systems. In order to achieve robustness in cloud computing, it is important to properly evaluate and respond to errors. One of the hardest parts of creating a fault-tolerant system is finding the source of the problem. Cloud computing has several advantages, including scalability, elasticity, high availability, and others. The cloudcomputing concept has revolutionised the IT sector since it offers numerous advantages to users on a personal, institutional, and national level. As many benefits as the cloud computing system offers, it is nevertheless vulnerable to malfunctions. Due to the massive nature of cloud computing, failures are unavoidable. To efficiently manage errors in a cloud setting, fault tolerance policies are often put into place. Fault tolerance methods aid in both preventing and dealing with system errors, whether they

originate in the hardware or the software. Applying fault tolerance strategies in the cloud is primarily motivated by the desire to improve availability, dependability, and recover from failure. In this overview paper, we looked at what cloud computing is, its parts, its service model, and its deployment models. The most severe drawback occurs when fault tolerance in one component reduces the efficiency of a dependent component. Eventually, the expenses will go up and the quality of the goods produced will suffer if faults are allowed.

REFERENCES

1. Bokhari, M.U., Shallal, Q.M., Tamandani, Y.K., 2016. Cloud computing service models: a comparative study. In: IEEE Int. Conf. Comput. Sustain. Glob. Dev. INDIACom, pp. 16–18.
2. M.K. Gokhroo, M.C. Govil, E.S. Pilli**Detecting and mitigating faults in cloud computing environment**IEEE Int. Conf. (2017)
3. M. Nazari Cheraghlou, A. Khadem-Zadeh, M. Haghparast**A survey of fault tolerance architecture in cloud computing**J. Netw. Comput. Appl., 61 (2016), pp. 81-92
4. Z. Amin, H. Singh, N. Sethi**Review on fault tolerance techniques in cloud computing**Int. J. Comput. Appl., 116 (18) (2015), pp. 11-17
5. Y.M. Essa**A survey of cloud computing fault tolerance: techniques and implementation**Int. J. Comput. Appl., 138 (13) (2016), p. 8887
6. T.J. Charity, G.C. Hua**Resource reliability using fault tolerance in cloud computing**Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on, IEEE (2016), pp. 65-71
7. Jialei Liu, I., Wang, Shangguang, Senior Member, IEEE, Zhou, Ao, Kumar, Sathish A.P., Yang, Fangchun, Senior Member, IEEE, Buyya, Rajkumar, Fellow, 2016. Using proactive fault-tolerance approach to enhance cloud service reliability. IEEE Trans. Cloud Comput., 1–1.
8. Zhang, P., Shu, S., Zhou, M., 2018b. An online fault detection model and strategies based on SVM-grid in clouds. IEEE/CAA J. Autom. Sin. 5 (2), 445–456
9. Wang, S. et al., 2016. Cloud Service Reliability Enhancement via Virtual Machine Placement Optimization Cloud, 10(June), 902–913.
10. Zhao, J., Xiang, Y., Lan, T., Huang, H.H., Subramaniam, S., 2017. Elastic reliability optimization through peer-to-peer checkpointing in cloud computing. IEEE Trans. Parallel Distrib. Syst. 28 (2), 491–502.
11. B. Hay, K. Nance, M. Bishop, Storm clouds rising: security challenges for IaaS cloud computing, in: 44th Hawaii International Conference on System Sciences (HICSS), IEEE, 2011, pp. 1–7.
12. L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A.V. Vasilakos, Security and privacy for storage and computation in cloud computing, Inform. Sci. 258 (2014) 371–386.
13. O.D. Alowolodu, B.K. Alese, A.O. Adetunmbi, O.S. Adewale, O.S. Ogundele, Elliptic curve cryptography for securing cloud computing applications, Int. J.Comput. Appl. 66 (2013).
14. M. Aslam, C. Gehrmann, M. Bjorkman, Security and trust preserving VM migrations in public clouds, in: IEEE 11th International Conference on Trust,Security and Privacy in Computing and Communications (TrustCom), 2012, pp. 869–876.
15. Y. Tang, P.P. Lee, J.C.S. Lui, R. Perlman, Secure overlay cloud storage with access control and assured deletion, IEEE Trans. Dependable Secure Comput.9 (6) (2012) 903–916.