A Review on Fault Tolerance in Cloud Computing

Ms. Reena Agnihotri (FEDA-E), GNA University, reena.agnihotri@gnauniversity.edu.in
Dr. Vikrant Sharma (Dean-FEDA-E), GNA University, dean.fet@gnauniversity.edu.in
Ms. Gurgeet kaur (FCS), GNA University, gurgeet.kaur@gnauniversity.edu.in

Abstract

On-premises and cloud solutions are always the process of discussion in any industry. Technology is changing day by day so organizations that have on premises solutions are shifting their platform to cloud solutions. Cloud computing can be defined as a model in which users can gain on-demand access to shared resources such as software, server storage, or any other hardware item over the internet or network. Cloud reduces the challenge for businesses and individuals of space considerations and insecurity of sensitive information. The hardware cannot be accessed directly by the consumer or user; it is only owned and managed by the provider. By using a browser, you can access assigned services and resources from anywhere. In this paper, I have explained the brief scrutiny of strategies for fault tolerance and caparisoning cloud computing.

Keywords: *SaaS, Proactive, Fault Tolerance, Reactive, Cloud Computing*

INTRODUCTION

Cloud computing technology ensures the availability of programmers and services that run on a dispersed network platform. Cloud computing refers to the collection networks, just as the term 'cloud' refers to the number of water molecules. Because networks in the form of the cloud handle the strain of service, this computing provides the best infrastructure other than premises on-premises. Some of the key features of cloud computing: Quality of services, high availability, low cost, flexibility for end-users etc. With the use of web connectivity, cloud computing provides consumers and organizations quick access to apps and resources without the need for installation and configuration on separate hardware. [1] Cloud computing provides three types of services: Software as a Service (SAAS), Platform as a Service (PAAS), and Information as a Service (IaaS) (IAAS). Every system's performance in cloud computing is crucial for reliability. Low service dependability can result in frequent breakdowns in cloud service performance, which costs the server money. It can also make users wait an excessive amount of time for services, which is referred to as tolerance to error.

In cloud computing, fault tolerance is a crucial issue, although fault tolerance techniques can help close the performance and reliability gap. Fault tolerance approaches [2] provide for failure recovery. So this paper is divided into a few sections: the origin of cloud computing, measurement parameters, fault tolerance policies, types, fault tolerance models and tools, challenges and comparative study,

ORIGIN OF CLOUD COMPUTING STACK

Cloud computing provides a centralized opportunity for users to get the IT services flexibly. It also balances the economic consumption of applications, data management and infrastructure [3]. Cloud computing offers a stack of services like SaaS, PaaS and IaaS. Here is the best example for the same:

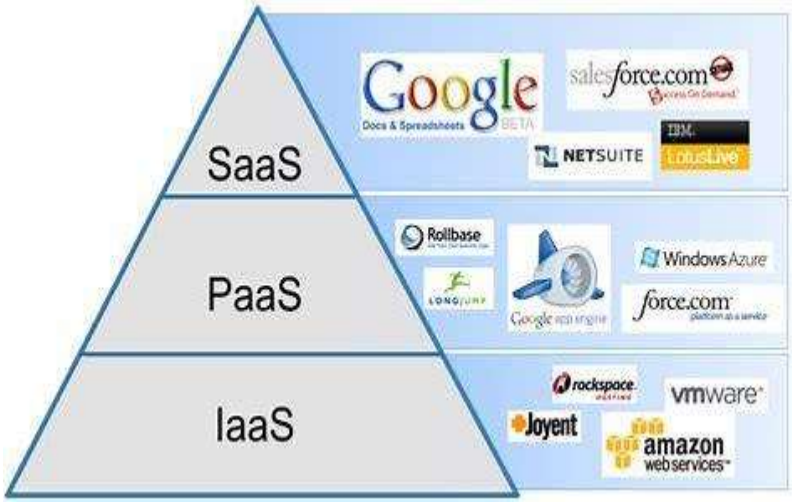

Fig 1.1 Cloud Computing Stack

3. FAULT TOLERANCE MEASUREMENT PARAMETERS

To assess the efficiency and effectiveness of cloud computing systems, numerous metrics are used to evaluate fault tolerance methodologies.

- Mean Time to Failure (MTTF): The average amount of time it takes for a system to fail once it has been operational.
- Mean Time to Fix - (MTTR): The average time required to repair an asset once it has failed.
- Mean Time between Failures (MTBF): The average time between failures is computed as follows:

  $$MTBF = MTTF + MTTR$$

- Reliability: System processing and running continuously without any failure.
- Availability: Probability that the system is performing without any bug at any given moment and is available to analyze the user task.

- Safety: When a system briefly fails to function properly, safety is the condition in which nothing happens.
- Maintainability: refers to how easy it is to repair a broken system. A system that is simple to maintain could also be very reliable.
- Adaptive: All processes are carried out automatically in response to specific conditions.
- Performance: Performance ensures the efficiency of the system.
- Response Time: How long an algorithm takes to respond.
- Throughput: measurement of how many tasks have been completed effectively.
- Usability: It refers to a user's ability to use an invention/product to achieve a goal with efficiency, effectiveness, and satisfaction.
- Overhead Associated: Calculate the overall overhead required in completing a task.
- Cost-effectiveness: It is a description of the system monetarily.

Table:1.1 Fault Tolerance Models Vs Fault Tolerance Parameters (Y -Yes, N- No, HG- High, AVG- Average, LW- Low)

Fault Tolerance Models/ Parameters	**AFTRC**	**LLFT**	**FTM**	**FTWS**	**CANDY**	**VEGA WARDEN**	**FT-CLOUD**	**MAGI CUBE**
Proactive	Y	N	N	N	N	Y	Y	Y
Reactive	N	Y	Y	Y	Y	Y	N	Y
Adaptive	Y	N	N	N	Y	N	Y	Y
Performance	HG	HG	AVG	AVG	AVG	HG	HG	HG
Response Time	AVG	AVG	AVG	AVG	AVG	HG	AVG	AVG
Scalability	HG	HG	LW	LW	HG	HG	HG	HG
Throughput	HG	AVG	AVG	AVG	HG	AVG	AVG	HG
Reliability	HG	HG	AVG	AVG	HG	HG	HG	HG
Availability	HG	HG	HG	HG	HG	HG	AVG	AVG
Usability	HG	AVG	AVG	AVG	AVG	HG	HG	HG
Overhead	AVG	LW	LW	LW	LW	HG	HG	AVG
Cost-effectiveness	AVG	LW	LW	LW	LW	LW	HG	HG

3. FAULT TOLERANCE POLICIES

Any cause of a system's failure is known as a Fault. Fault tolerance provides a kind of specialty in a system that prevents a cause of hardware or software failure [4]. It contains proactive steps to prevent such errors and bugs. This technique is capable of providing the services in case more and more failures occur frequently and it also maintains the availability and reliability of the system. Two solutions to this problem: Fault detection and Fault repair.

Reactive Fault Tolerance

When there are a lot of failures, this strategy is applied. The optimal policy for reactive fault tolerance is checkpoints and restart. Different tasks are operating on distinct resources to ensure that the duplicate task does not crash. It can be done with techniques like HA proxy, Hadoop, and others [5]. Another feature is job migration, which allows you to move the job to another machine to avoid any delays in the event of a failure. Rollback is also significant since it gives additional resources in the event of a single process failure. S-Guard is another name for it. Rescue Workflow is a typical strategy that allows the workflow to continue even if one or more tasks fail, making it impossible to move on to the next destination without catering to the failed one. User-defined exception handling occurs when a user sets the specific treatment for a task failure in a workflow.

Proactive Fault Tolerance

By predicting any suspect in a present process and replacing them with other functional components, these rules eliminate recovery procedures from faults, errors, and defects. These approaches are likewise based on self-healing principles. The divide and conquer strategy is used to break down a major task into smaller portions. This partition was created primarily to increase the system's performance. Pre-emptive migration requires a feedback loop control approach because an application is constantly examined and analyzed.

Balancing the load, when the memory and CPU load surpasses a maximum/certain limit, this method is utilized to balance the load. The load of an overloaded CPU is transferred to another CPU that is not overloaded.

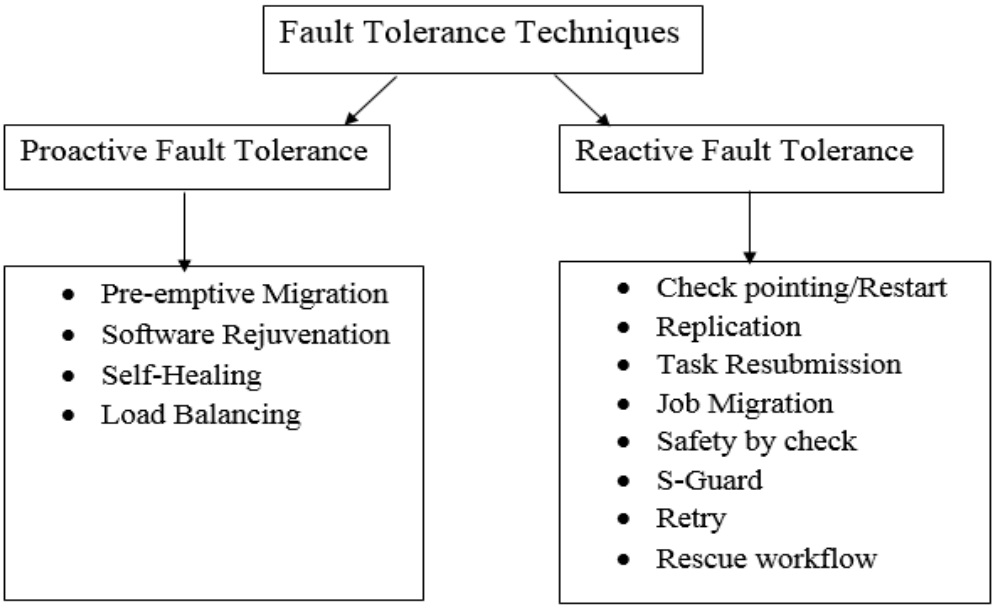

Fig 1.2 Classification of Fault Tolerance techniques

4. TYPES OF FAULT TOLERANCE

Hardware fault tolerance

The physical fault is also known as the hardware fault of any system. Whether it is a backup failure, system crash, automatic shutdown, update failure, windows server fault etc. Error handling and dynamic recovery are the best examples of hardware fault tolerance. In dynamic retrieval, a copy of work and calculation is made to run in a real time real-time. This process is also known as self-repair. [6]

Build-in self-test feature (BIST)

This method ensures that the system can run tests at predetermined intervals to look for fault propagation. When a mistake occurs, it is set up to replace the faulty part and transition to the redundant mode.

Triple Modular Redundancy (TMR)

Three redundant copies of a fault component are processed and run concurrently in this technique. Performance-based selection is used to address a single defect at a time.

Circuit Breaker (CB)

This is a circuit-based design that allows for circuit breaking in distributed systems to avoid catastrophic failures.

Software - fault tolerance

N Version Programming

This method refers to n developers creating limitless variations of a program. All of these copies are made simultaneously, with the highest level of fault tolerance selected. This is a software fault-detection approach used during one of the SDLC stages. It is selected and utilized to handle a single fault at a time.

Recovery Blocks

This method is similar to the previous one, except that the redundant copies are not run at the same time [7]. They are generated one by one using a distinct set of algorithms.

When task deadlines exceed computing time, this strategy is employed.

Few Techniques in Fault Tolerance:

It's a technique for detecting node failures and crashes in a given process. Based on the results it delivers, a failure detector can be characterized as trustworthy or unreliable. When a failure detector's output is consistently accurate, it is referred to as a dependable failure detector. A non-reliable failure detector provides information that is likely to be inaccurate, takes a long time for any defective process to complete, and produces erroneous results by suspecting processes that have not crashed [10]. The majority of failure detectors come under this group. Failure detectors have a few correctness properties:

- Completeness: When a process fails, it is explicitly identified by at least one other process that is not defective. Completeness also refers to the failure detector's ability to permanently suspect every failed process in the database.
- Accuracy: There are no error failure detections, which means that if a process is set to failed fail, it has failed. Low false positives lead to high accuracy [11]. It is impossible to develop a failure detector that is 100 percent accurate and provides the whole result over a realistic network. Completeness is guaranteed by real-world failure detectors, while accuracy is either partial or probabilistic. Accuracy and completeness can be compromised.

5. FAULT TOLERANCE MODELS

AFTRC – Adaptive fault tolerance is the term for this. Real time computing model and is capable for real time application which has a high processing structure in the environment

of\scloud computing. In this paradigm, the system detects the defect in advance and makes a decision based on the processing nodes' reliability. [20]

LLFT : This is known as a low latency fault tolerance approach, and it is a middleware for providing fault tolerance in cloud-based sharing applications. The data replication procedure in this middleware technique provides fault tolerance. To guard against various forms of errors, the programme employs a semi-passive or semi-active replication procedure. [21]

FTM: The fault tolerance approach is expected to solve previous methodologies' limitations in data computing for on-demand services. This architecture assures resilience and dependability by employing cutting-edge technique that allows the user to counter and apply the necessary level of fault tolerance without having to understand how it works. It can be seen of as a collection of various web services components, each with its own set of capabilities [22]. It's a type of reactive architecture that uses three fault tolerance methods: replication, checkpoint/restart, and task migration.

FTWS: The fault tolerance workflow scheduling model includes a scheduling technique that uses resubmission and duplication of tasks based on task priority in a heuristic matrix approach to offer fault tolerance [23]. This paradigm is built on workflow, which is a collection of actions that are performed in a specific order depending on data and control dependencies. In the cloud, task failure is taken into account while scheduling process. FTWS duplicates and schedules jobs to ensure that they are completed on time.

Candy - Candy is a component-based availability modelling framework that builds a full availability model semi-automatically from system specifications expressed in systems modelling language. [24] This approach is based on the fact that one of the key characteristics of cloud services is high availability guarantee, which is also one of the most significant and tough concerns for cloud service providers.

Vega-warden This is a one-of-a-kind user management solution that aids in the provisioning of a global user space for various virtual infrastructure and application services in the cloud computing environment. This paradigm is widely utilized in virtual cluster-based cloud computing environments to address security and usability issues that occur from infrastructure sharing [25].

FT-Cloud: This is a component-based framework whose architecture is used to create cloud applications. To identify the component, FT-Cloud uses the component invocation structure and frequency. To automatically determine fault tolerance levels, a particular method was developed [26].

Magi-Cube: This is a very reliable and low-redundancy storage solution for cloud computing. They utilize HDFS's architecture as a file read/write and metadata management storage system as they build the system on top of it. They also rebuilt a file script and repair component that runs in the background, providing high performance and dependability at a minimal cost. [27]

6. TOOLS USED FOR IMPLEMENTING FAULT TOLERANCE

• HAProxy -The first type of reactive architecture is the fault tolerance model. Replication and job migration techniques are utilized in this. There are two server machines in operation, as well as one HAProxy in monitoring mode. The replica of one server machine is on a second server machine. If the first server machine fails, the second server machine will take over and handle the error on its own. It's used in the cloud to handle server failover [29].

• SHelp: It's a lightweight runtime system for virtual machines that can withstand software faults. It can also be used in a cloud environment to implement checkpoints.

ASSURE: It benefits rescue points for dealing with programmer expected faults in a cloud environment [30].

• Hadoop: It can be used to develop fault tolerance plans for the cloud because it is made for data-intensive applications.

• Amazon Elastic Compute Cloud (EC2): For fault tolerance, it defines a virtual computing environment to run Linux-based applications [31].

Table 1.2 Comparative study of fault tolerance models:

Sr. No.	Technique	Type	Tools	Implementation Environment
1	Pre-emptive Migration	Proactive	HAProxy	VM
2	Software Rejuvenation	Proactive	Assure	VM

3	Self-Healing	Proactive	HAProxy/Assure	VM
4	Check pointing/Restart	Reactive	SHelp/Assure	VM
5	Replication	Reactive	HAProxy/ AmazonEC2	Cloud
6	Task Resubmission	Reactive	AmazonEC2	Cloud
7	Job Migration	Reactive	HAProxy/ Hadoop	VM/ Cloud
8	S-Guard	Reactive	AmazonEC2/Hadoop	Cloud
9	Rescue workflow	Reactive	Hadoop	Cloud

7.CHALLENGES OF IMPLEMENTING FAULT TOLERANCE

Because fault tolerance has a complicated and interdependent structure, it necessitates considerable thought and study. For many instances of an application running on separate virtual machines, autonomic fault tolerance methodology is required. To manage all forms of failures, algorithms must be centralized, which is nearly difficult, which is why engineers use a benchmark-based methodology to compare the performance of fault tolerance components to similar ones. When developing a fault tolerance system, all services must prioritize vital tasks. Because it powers numerous other units, the DB Team must be given special consideration.
The company will have to work on the trial test after formalizing the priorities. For instance, the company must have a forum website where users may log in and leave comments. When the authenticated service fails, the user will be unable to login. The system then becomes read-only and no longer serves its purpose. However, with fault-tolerant systems, remediation will be assured, and the user will be able to search for information with minimal disruption.

The notions of redundancy and replication define fault tolerance in None Point Failure, although with some minor consequences. The system is not fault-tolerant if there is not even a single point of failure. The graphic below depicts the current state of Fault Tolerance implementation.

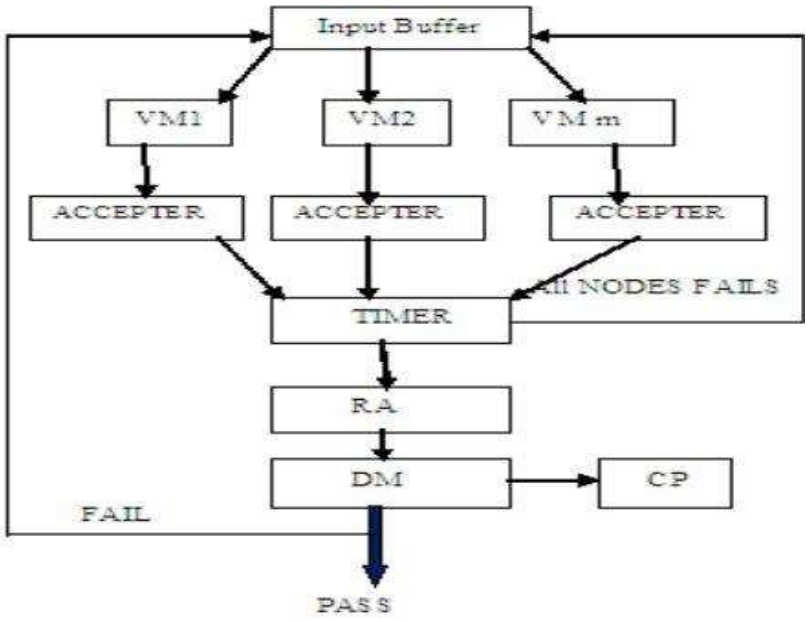

Fig 1.3 Methodology of implementation in fault tolerance

8. CONCLUSION

The presented study examined fault tolerance strategies in cloud computing, including research problems and tools for implementing fault tolerance techniques. Fault tolerance is essentially about tolerating, avoiding, and preventing defects and errors that persist in a system after it has been developed and implemented. This paper compared several fault tolerance approaches as well as fault types and tolerance techniques. The tools for fault tolerance were also explored in this paper, as well as their comparison. Failures in virtual machines and changes in the cloud environment caused by failures will be detected in the future by methods that may be offered to eliminate the problem. Various fault tolerance strategies, models, and algorithms to improve cloud service reliability have been described. We have proposed several future research objectives based on the limits of present fault tolerance approaches and upcoming technology in related sectors.

REFERENCES

1 Antonina Litvinova, Christian Engelmann and Stephen L. Scott (2009), A Proactive Fault Tolerance Framework for High-Performance Computing,

2. Golam Moktader Nayeem, Mohammad Jahangir Alam (2006), Analysis of Different Software Fault Tolerance Techniques,

3 Steven Y. Ko, Imranul Hoque, Brian Cho and Indranil Gupta (2010) ,On Availability of Intermediate Data in Cloud Computations,

4. Manish Pokharel and Jong Sou Park (2010) ,Increasing System Fault Tolerance with Software Rejuvenation in E-government System", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5.

5 L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner (2008), A break in the clouds: towards a cloud definition," SIGCOMM Computer Communication Review,vol. 39, pp. 50–55.

6. Imad M. Abbadi (2010), Self-Managed Services Conceptual Model in Trustworthy Clouds' Infrastructure,

7. By . Z. Amin, H. Singh, N. Sethi (2015), Review on fault tolerance techniques in cloud computing Int. J. Comput. Appl., 116 (18), pp. 11-17.

8. Salma M. A. Ataallah, Prof. Salwa M. Nassar, Prof. Elsayed E. Hemayed (2015), Fault Tolerance in Cloud Computing – Survey, IEEE.

9. " Survey on Fault Tolerance Techniques in Cloud Computing Environment", *International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-9, December 2015 ISSN: 2395-3470, by* V.M.Sivagami, Dr. K.S.EaswaraKumar.

10. " A Survey of Fault Tolerance in Cloud Computing", *International Journal of Arts, Science and Humanities*, Volume : 6 , Special Issue : 1 , September ,2018 by R.Archana.

11. " Trust And Fault Tolerance Models In Cloud Computing: A Review ",INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 8, ISSUE 11, NOVEMBER 2019, Shivani Jaswal, Manisha Malhotra.

12. A Survey of Fault-tolerance in Cloud Computing: Concepts and Practice, Research Journal of Applied Sciences, Engineering and Technology 11(12): 1365-1377, 2015

Ameen Alkasem and Hongwei Liu,

13. , "An Efficient Fault Tolerance Mechanism Based on Moving Averages Algorithm" © 2013, IJARCSSE, ISSN: 2277 128X by Amritpal Singh, Supriya Kinger.

14 Bala, A., &Chana, I. (2012). Fault tolerance-challenges, techniques and implementation in cloud computing. International Journal of Computer Science Issues (IJCSI), 9(1), 288.

15. Zhang, Y., Zheng, Z., &Lyu, M. R. (2011, July). BFTCloud: A byzantine fault tolerance

framework for voluntary-resource cloud computing. In 2011 IEEE 4th International Conference on Cloud Computing (pp. 444-451).IEEE.

16. Kaur, J., &Kinger, S. (2013). Analysis of different techniques used for fault tolerance. IJCSIT Int. J. Comput.Technol, 4, 737-741.

17 Lakshmi, S. S. (2013). Fault tolerance in cloud computing. Int. J. Eng. Sci. Res. IJESR, 4, 1285-1288.

18 Lim, J., Chung, K. S., Gil, J. M., Suh, T., & Yu, H. (2013, February). An unstructured termination detection algorithm using gossip in cloud computing environments.In International Conference on Architecture of Computing Systems (pp. 1-12).Springer, Berlin, Heidelberg.

19 Egwutuoha, I. P., Chen, S., Levy, D., &Selic, B. (2012, May). A fault tolerance framework for high performance computing in cloud.In 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012) (pp. 709 -710).IEEE.

[20] Sheheryar MalikandFabriceHuet (2011), Adaptive Fault Tolerance in Real Time Cloud Computing, IEEE World Congress on Service

[21] Wenbing Zhao, P.M. Melliar and L.E. Mose, (2010), Fault Tolerance Middleware for Cloud Computing" IEEE 3rd International Conference on Cloud Computing

[22] Ravi Jhawar, Vincenzo Piuri and Marco Santambrogio"A Comprehensive Conceptual System level Approach to Fault Tolerance in Cloud Computing" IEEE

[22] Jayadivya S K, JayaNirmala S, Mary SairaBhanus"Fault Tolerance Workflow Scheduling Based on Replication and Resubmission of Tasks in Cloud Computing" International Journal on Computer Science and Engineering (IJCSE)

[23] Fumio Machida, Ermeson Andrade, Dong SeongKim and Kishor S. Trivedi"Candy: Component-based Availability Modeling Framework for Cloud Service Management Using Sys-ML" 2011 30th IEEE International Symposium on Reliable Distributed Systems.

[24] Jianlin, Xiaoyi Lu, Lin Yu, YongqiangZou and Li Zha" (2010), Vega Warden: A Uniform User Management System for Cloud Applications, Fifth IEEE International Conference on Networking, Architecture, and Storage.

[25] ZibinZheng, Tom Chao Zhou, Michel R. Lyu, and Irwin king 2010 ,FT-Cloud: A Component Ranking Framework for Fault-Tolerant Cloud Applications , IEEE 21st International Symposium on Software Reliability Engineering.

[26] QingqingFeng, Jizhong Han, Yun Gao, Dan Meng (2012) , Magicube: High Reliability and Low Redundancy Storage Architecture for Cloud Computing IEEE Seventh International Conference on Networking, Architecture, and Storage.

[27] http://haproxy.1wt.eu/download/1.3/doc/configuration.txt.

[28] Gang Chen, Hai Jin, Deqing Zou, Bing Bing Zhou, Weizhong Qiang, Gang Hu (2010), "SHelp: Automatic Selfhealing for Multiple Application Instances in a Virtual Machine Environment", IEEE International Conference on Cluster Computing.

[29] S. Sidiroglou, O. Laadan, C. Perez, N. Viennot, J. Nieh, and A. D. Keromytis,

"ASSURE: Automatic Software Self-healing Using REscue points", Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating System.

[31] AmazonElasticComputeCloud(EC2) HYPERLINK "http://www.amazon.com/ec2/" http://www.amazon.com/ec2/

[32] Manoj Kumar Malik, Ajit Singh, Abhishek Swaroop (2019), Fault Tolerance in Cloud Computing: A Major Research Challenge International Journal of Innovations in Engineering and Technology (IJIET), Volume 14 Issue 4.

[33] Ameen Alkasem and Hongwei Liu (2015), A Survey of Fault-tolerance in Cloud Computing: Concepts and Practice, Research Journal of Applied Sciences, Engineering and Technology 11(12): 1365-1377.

[34] Priti Kumari, Parmeet Kaur (2018),A Survey of Fault Tolerance in Cloud Computing, Journal of King Saud University - Computer and Information Sciences Vol 8, PP 33.