EQUIVALENCE DATA GENERATION TECHNIQUE IN CASE OF DATA LOSS DUE TO FAULT OCCURRENCE IN VIRTUAL MACHINES USED IN CLOUD COMPUTING

Umesh Dwivedi, PhD Scholar, Uttar Pradesh Technical University, Uttar Pradesh, India.
Asst. Professor, Department of Computer Application, Shri Ramswaroop Memorial Group of Professional Colleges, Lucknow, India. Email: umesh2112@gmail.com
Dr. Harsh Dev, Professor and Dean in Research Wing, Pranveer Singh Institute of Technology, Kanpur, India. Email:drharshdev@gmail.com

ABSTRACT:

Cloud computing is a computing like a blessed child in the field of computer. It is able to provide the solution of any problem in computers. As the data and user requirement increased two new problem came into existence, one is to store this huge amount of data and another is how to fulfil all the requirements of an individual user. As we know, computers have limitations in both the aspects, in storage as well as in the individual requirements . Ahead of this, user wants to fulfil their requirements round the clock and upto endless time. Both of these problems are solved by the cloud computing. Cloud gives unlimited space and side by side no requirement of new software or application installation on their own machine. Thats why cloud is said to be the solution of every problem in the field of computer. But if any fault occurs,the whole architecture of cloud gets crippled.. To solve this problem many fault tolerant techniques are being proposed by the researchers in the field of cloud computing. Many of them are very useful. In the same sequence, we have proposed a new fault tolerant technique named as equivalence data generation technique to regenerate the lost or crashed data of any virtual machine which is being used by cloud architecture. We analysed this technique on four virtual machine architecture and successfully recovered the data of the lost virtual machine.

Index Terms: Cloud computing, Fault tolerance, hardware failure, Redundancy, equivalence.

I. INTRODUCTION:

Cloud computing is a new paradigm in distributed computing. With the help of virtualization it provides a very useful service oriented architecture[1,2]. Because of its massive utility world is diverting towards this architecture slowly but steadily. Cloud computing has great ability to connect the Internet and any type of network for resource utilization and provides these resources on the payment basis[3][4]. Because of extensive use of cloud computing there is a need of fault free system. So the fault tolerance becomes the key factor to be considered.

Fault tolerance considers each and every technique necessary for a robust and fault free system[5]. Delivery of correct service in adverse situation is the basic outcome required from the fault tolerant system.[7] Faults are unavoidable. Hence the situation can be handled more accurately by restricting the use of system in adverse situation not at the time of occurrence of fault. Such systems are said to be robust systems. In the case of grid computing where there is sharing between computational power, data, storage and resources fault tolerance is more important because the impact of fault is more in grid because of distributed resources.[8]

Reliability of system means that the system can run in any condition without any hindrance. Availability means that the system is ready to use any time. Fault tolerance techniques are used to increase reliability and availability.[7] .For designing the fault tolerant system first we have to design a system which can detect the fault.

Fault Detection system: There are two main services for detection of fault. i.e. Pull Model and Push Model.[9] In Pull Model different grid components are responsible for sending the signals for fault detector and in Push model fault detector is responsible for sending period signals to different grids to check their current situation.

This paper intends to give a superior comprehension of adaptation to non-critical failure challenges and distinguishes different devices and procedures utilized for adaptation to non-critical failure. With the assistance of virtualization method various examples of an application can keep running on a few virtual machines. On the off chance that any one server goes down, there turns into a need to execute an autonomic adaptation to non-critical failure method that can deal with these sorts of shortcomings. To address this issue, cloud virtualized framework design has been proposed and actualized utilizing HAProxy. The proposed

Received: 15 May 2019/Accepted 05 July 2019.

design likewise has been approved through exploratory outcomes.

The rest of the paper is organized as follows. Section 2 discusses different fault tolerant strategies. Section 3 explains different fault tolerance design techniques based on their policies. Section 4 presents practical challenges while implementing fault tolerance in cloud computing. Section 5 explains the methodology used in this paper for creating fault tolerant technique. Section 6 explains the algorithm designed by us. Section 7 explains the result after implementing our prescribed method. Section 8 finally concludes the paper. Section 9 prescribes some future scope of this research.

II. FAULT TOLERANCE DESIGN TECHNIQUES:

The key ingredients in the case of fault tolerance are Error Processing and Fault treatment. In the case of error processing error removal is done before the occurrence of failure and in the fault treatment option faults are avoided to be activated again.

Error processing can be classified in three stages: (i) Error detection: In this state identification of erroneous states is done. (ii) Error diagnosis: In this state Damage assessment is done. (iii) Error recovery: Finally in this state Error free state is substitute to erroneous state.

This error recovery state can be further divided in two categories: (i) Back word Recovery: Here system brought back to the previously visited state with the help of check pointing mechanism. (ii) Forward Recovery: Here erroneous state is discarded and correct state is designed without loosing the computation.

In the next state Fault treatment is done. This fault treatment step can be classified in two steps: (i) Fault diagnosis: This stage determines error causes. (ii) Fault isolation: This stage removes faulty components from subsequent execution process. But in this case system is no longer available for delivering the same service. Modification of system structure is done and non failed components delivers degraded service.

III. FAULT TOLERANT STRATEGIES:

Adaptation to internal failure in PC framework is accomplished through repetition in equipment, programming, data, or potentially time. Such excess can be executed in static, dynamic, or half and half setups.

Adaptation to internal failure can be accomplished by numerous systems, for example, deficiency concealing, reconfiguration, check pointing, Job movement, task repetition and so on.

Fault reconfiguration approaches are fault detection, fault location, fault containment, fault recovery.

The concept of redundancy works on the concept of expansion of data, assets or time past what is required for typical task.

Equipment repetition is the expansion of additional equipment, for the reason either distinguishing or enduring issues. Programming repetition is the expansion of additional product, past what is expected to play out an offered capacity, to distinguish and perhaps endure issues.

Data repetition is the expansion of additional data past that required to actualize a given capacity; for instance, blunder discovery codes.

Time excess uses extra time to play out the elements of a framework with the end goal that deficiency identification and frequently adaptation to internal failure can be accomplished. Transient deficiencies are endured by this strategy.

The utilization of repetition can give extra capacities inside a framework. Be that as it may, excess can have significant effect on a framework's exhibition, size, weight and power utilization. Cloud computing data canters hosts hundreds of thousands of servers. As the number of servers increases with the increasing demand of cloud, server failure chances also increases.

IV. PRACTICAL CHALLENGES WHILE ACTUALIZING FAULT TOLERANCE IN CLOUD COMPUTING:

While implementing he adaptation to non-critical failure in cloud computing there are many practical challenges as follows.

- ✓ Cloud architecture is distributed architecture so it becomes difficult to handle the implemented fault tolerance technique[6].

- ✓ Any implemented technique has to handle the interdependent issue with the multiple instances of an application running on several virtual machines.

- ✓ There is always a need of automatic fault tolerance technique with each virtual machine instance in cloud.[11]

- ✓ For implementing reliable system different technologies of different contending

Received: 15 May 2019/Accepted 05 July 2019.

sellers of cloud foundation are should be incorporated.[12]

✓ There is always a need of new technique to be integrated with current working fault tolerance techniques.[13]

✓ To ensure high reliability and availability with multiple cloud computing providers there is a need of independent software stack. [14][15].

V. METHODOLOGY:

Proposed methodology begins with the utilization of virtual Machines. But this utilization get hamper if any machine fails. This failure is because of any reason. Reason may be hardware failure, software failure or even network failure. In current paper we are going to consider the hardware failure situation. Hardware failure may be of many kinds but mainly and biggest effecting failure can be data loss. Biggest reason of data failure is disk failure. But this situation can not be avoided. So we have to pay special attention for this situation.

Proposed methodology handles the hare ware failure. In hardware failure storage disk failure is a vary common problem. In that condition major data loss occurs because of which client receives no data or wrong data. Situation can be handle by storing same data on many disks. But this will create data redundancy and redundancy is a big problem itself. Because it not only consumes multiple storage spaces but also it also creates data updation very difficult. No other way is gives very fruitful results. In this paper we proposed the technique named as equivalence technique. This technique is used here for retrieving the lost data due to any disk failure.

VI. PROPOSED ALGORITHM:

Only one additional disk is required in our technique. This disk drive is used to store Equivalence data which is used in the fault tolerance techniques. This disk storage is to hold the Equivalence bits, a mathematical construct that allows re-creation of the missing data. Calculation of Equivalence is a function of our controller.
Equivalence computations are utilized for adaptation to internal failure by ascertaining the information in two drives and putting away the outcomes on a third. The Equivalence is figured by XOR'ing a bit from drive 1 with a bit from drive 2 and putting away the outcome on drive 3 After a bombed drive is supplanted, the RAID controller revamps the lost information from the other two drives. Strike frameworks frequently have a "hot" save drive prepared and hanging tight to supplant a drive that fizzles.

Working method of Equivalence technique is as follows:

Step 1: Let we have 4 disks then there must be an another disk to store Equivalence data. In total there will be 5 disks.

Step 2: Let us assume any one disk fails missing data an be recalculated in the next step.

Step 3: Missing value is calculated by subtracting the sum of the rest of the elements stored in different disks other from the failed disk from the Equivalence value.

Step 4: Restoring the calculated data in new disk and connecting the disk to the network.

Step 5: End

Identicalness data can be put away on independent, devoted circle drives or dispersed over every one of the drives. On the off chance that one of the information plates comes up short, the missing worth can be determined by subtracting the aggregate of the remainder of the components from the Equivalence esteem.
Procedure can be understood by the following flow chart.

Received: 15 May 2019/Accepted 05 July 2019.

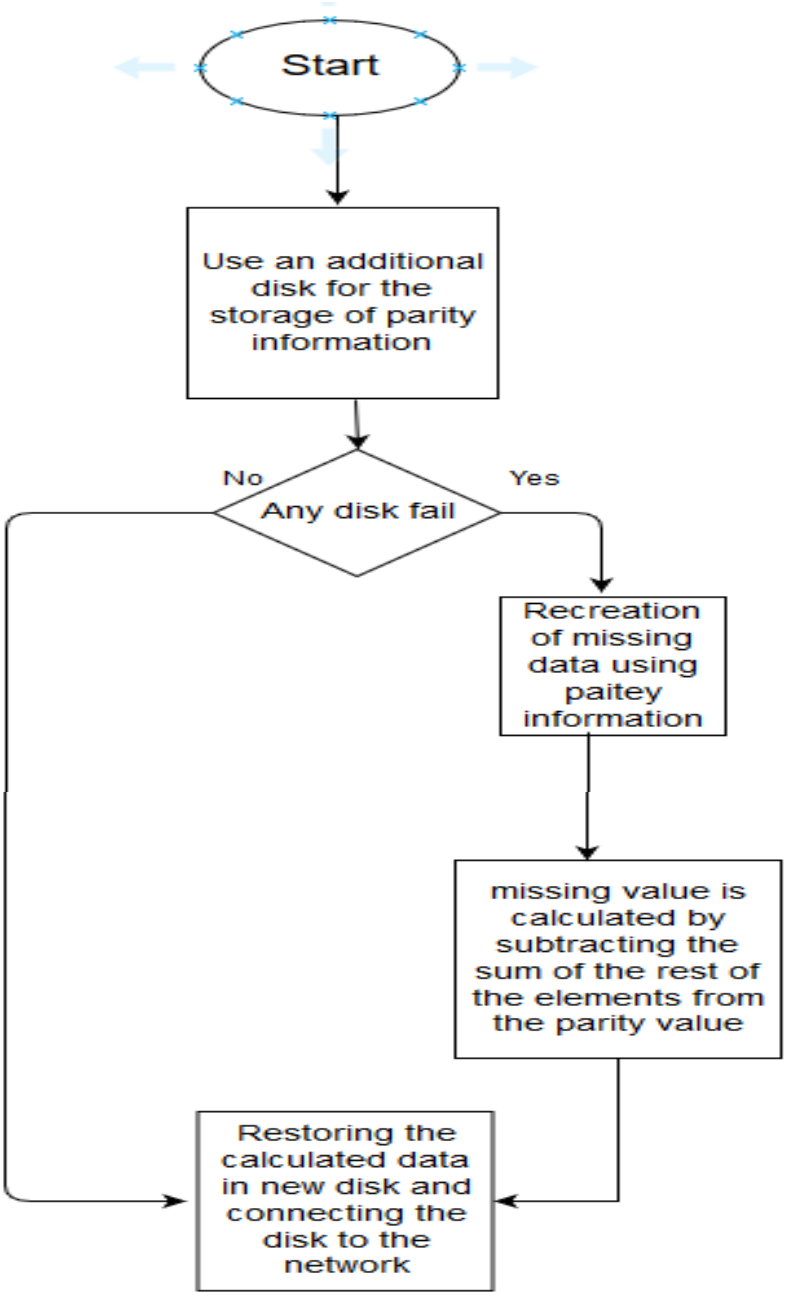

Figure 1: Flow Chart

Figure 1 demonstrate the process flow of our proposed fault tolerance model. In which it is shown that we use an additional disk for storage of Equivalence information. In case of any disk fails recreation of missing data is done by the Equivalence information. Missing worth is determined by subtracting the whole of the remainder of the components from the Equivalence value. After this step restoring the calculated data in new disk and connecting the disk to the network.

VII. RESULT ANALYSIS:

To study the effectiveness of proposed data generation algorithm we have taken the data of four virtual machines situated in a data centre which can be used by different

Received: 15 May 2019/Accepted 05 July 2019.

clients. Virtual machines are with different bandwidth and processing speed. We provided this same data to two other data centres having four-four virtual machines each with same capacity as in our sample data centre.

In our experiment we assumed that we have four virtual machines having different data for the cloud use. An extra virtual machine is also being used in the data centre for the storage some extra information about the data stored in four virtual machines. This extra information will be used to regenerate the crashed data of any one virtual machine.

In our research we have taken four virtual machines each having 5 different locations. Each location has 5 different slots and data is stored in each slot in binary format.

Arrangement is as follows.

There are four Virtual Machines VM1, VM2 VM3 and VM4..

Virtual Machine 1

Location 1	Location 2	Location 3	Location 4	Location 5
11010	10101	11011	11101	10101
10110	11001	10100	11011	11011
11001	11100	11001	10001	11000
10001	11100	11001	11000	10011
10110	11011	11100	11000	11100

Virtual Machine 2

Location 1	Location 2	Location 3	Location 4	Location 5
10110	10001	10010	11001	10001
11100	11001	10011	11001	10011
11000	10100	11001	10001	10000
11100	11100	11000	10001	10011
11000	11100	11100	10011	10001

Virtual Machine 3

Location 1	Location 2	Location 3	Location 4	Location 5
11011	10101	11011	11101	10101
10110	11001	10100	11011	11011
11101	11100	11001	10001	11000
10001	11100	11001	11000	10011
11110	10011	11100	11000	11100

Received: 15 May 2019/Accepted 05 July 2019.

Virtual Machine 4

Location 1	Location 2	Location 3	Location 4	Location 5
10110	10001	10011	11001	10001
11100	11001	10011	11001	10011
11000	10100	11001	10001	10000
11100	11100	11000	10001	10011
10000	11100	11111	11011	10001

To make these virtual machines fault tolerant we used one extra disk called Equivalence disk.

In our experiment Equivalence disk stores the sum of individual location data of each virtual machine in its individual location. Every location has five slots and each slot is of five bits. We have stored the sum of individual bits of each slot at each location in individual bit position of each location of Equivalence disk. For example at location 1 of Equivalence disk has bit location. At position one of location 1 of Equivalence disk we store the sum of location 1 of each slot in each virtual machine. Such as

P Disk Location 1, Bit1 position =
VM1 Location 1, Slot 1 + VM1 Location 1, Slot 2+VM1 Location 1, Slot 3 + VM1 Location 1, Slot 4+VM1 Location 1, Slot 5 +
VM2 Location 1, Slot 1+ VM2 Location 1, Slot 2+VM2 Location 1, Slot 3+ VM2 Location 1, Slot 4+VM2 Location 1, Slot 5 +
VM3 Location 1, Slot 1 + VM3 Location 1, Slot 2+ VM3 Location 1, Slot 3+ VM3 Location 1, Slot 4+ VM3 Location 1, Slot 5+
VM4 Location 1, Slot 1+VM4 Location 1, Slot 2+VM4 Location 1, Slot 3+VM4 Location 1, Slot 4+VM4 Location 1, Slot 5

Similarly P Disk Location 1, Bit 2 position, P Disk Location 1, Bit 3 position, P Disk Location 1, Bit 4 position, P Disk Location 1, Bit 5 position data can be calculated.

In the similar fashion Location 2, Location 3, Location 4 and Location 5 data can also be calculated.

Final calculated data in Equivalence disk is as follows.

Equivalence disk

location 1	location 2	location 3	location 4	location 5
42341	40204	42043	44204	40204
42420	44004	40204	44024	42044
44102	42400	44004	40004	42000
42202	44400	44002	42002	40044
42220	43222	44401	44022	42204

After creation this Equivalence disk with filled data nothing has to be done.

In case of any virtual machine failure its data can be calculated as follows.
Location 1, slot 1data of disk 3= Location 1, slot 1 data of p disk -
(

In our experiment we assumed that VM 3 fails. To restore the data of VM 3 calculation is as follows:

VM1, Location 1, slot 1 + VM1, Location 1, slot 2 + VM1, Location 1, slot 3 + VM1, Location 1, slot 4 +VM1, Location 1, slot 5 +

Received: 15 May 2019/Accepted 05 July 2019.

VM2, Location 1, slot 1 + VM2, Location 1, slot 2 + VM2, Location 1, slot 3 + VM2, Location 1, slot 4 +VM2, Location 1, slot 5+ VM4, Location 1, slot 1 + VM4, Location 1, slot 2 + VM4, Location 1, slot 3 + VM4, Location 1, slot 4 +VM4, Location 1, slot 5)

VM3, Location 1, slot 1 + VM3, Location 1, slot 2 + VM3, Location 1, slot 3 + VM3, Location 1, slot 4 +VM3, Location 1, slot 5+
In the similar fashion Location 2 with all slots, Location 3 with all slots, Location 4 with all slots and Location 5 with all slots data is calculated.

We get the result =

Final calculated data is :

Equivalence Disk		VM 1		VM 2		VM 4		
location 1		location 1		location 1		location 1		
42341		11010		10110		10110		11011
42420		10110		11100		11100		10100
44102	-	11001	+	11000	+	11000	=	11001
42202		10001		11100		11100		11001
42220		10110		11000		10000		11100

This is the calculated data of VM3 which was crashed. This data can be replaced with the crashed Virtual Machine data.

VIII. CONCLUSION:

As the system stops working means system is not fault tolerant for that particular fault which occurred. To avoid this abnormal system failure condition we have to use some some fault tolerant technique. Fault tolerant techniques are of two type. Proactive technique which prevent fault before it occurs. Other is reactive fault tolerant technique which handles fault after its occurrence. This paper provides a reactive fault tolerant technique which comes into action after its occurrence. We proposed a fault tolerant technique which is fully error free and successful in generating the lost data of any virtual machine. After data generation this data can be replaced with the faulty data or crashed data virtual machine. So the cloud services can be provided without any break.

IX. FUTURE SCOPE:

Our proposed technique can be implemented not only on the binary stored data but it is applicable to the octal and Hexadecimal formats. This technique in future can be further be enhanced not only for storage failure but also for any other type of failure in virtual machines such as network failure, task failure or failure of any process because data is required in all the processes and data can be handled by our proposed technique. In future we will try to design a model to handle any fault in the cloud services.

X. REFERENCES:

[1] M. A. Vouk, "Cloud Computing – Issues, Research and Implementations", Department of Computer Science, North Carolina State University, Raleigh, North Carolina, USA, Journal of Computing

Received: 15 May 2019/Accepted 05 July 2019.

and Information Technology - CIT 16, vol. 4, (2008), pp. 235–246.

[2] B. S. Taheri, M. G. Arani and M. Maeen, "ACCFLA: Access Control in Cloud Federation using Learning Automata", International Journal of Computer Applications, vol. 107, no. 6, (2014), pp. 30-40.

[3] Sun Microsystems, Inc., "Introduction to Cloud Computing Architecture", White Paper 1st Edition, (2009).

[4] M. Fallah, M. G. Arani and M. Maeen, "NASLA: Novel Auto Scaling Approach based on Learning Automata for Web Application in Cloud Computing Environment", International Journal of Computer Applications, vol. 113, no. 2, (2015), pp. 18-23.

[5] Seyyed Mansur Hosseini and Mostafa Ghobaei Arani, "Fault Tolerance Techniques in Cloud Storage: A Survey", International Journal of Database Theory and Application, Vol 8, No. 4 (2015), pp. 183-190.

[6] A bala and I chana, "Fault tolerance-Challenges, Techniques and Implementation in Cloud Computing". IJCSI International Journal of computer Science Issues, Vol 9, No. 1 (2012), pp. 1694-0814.

[7] Benjamin Lussier, A Lampe, R Chatila, J. Guiochet, F. Ingrand,M. O. Killijian and D Powell, "Fault Tolerance in Autonomus Systems: How and How much?", LAAS-CNRS 7 Avenue du Colonel Roche, F-31077, Toulouse Cedex 04, France.

[8] P Latchoumy and S A Khader, "Survey on Fault tolerance in Grid Computing", IJCSI International Journal of Computer Science, Vol 2, No. 4, (2011).

[9] Nazir, B. and Khan, T.,(2006), "Fault Tolerant Job Scheduling in Computational Grid Emerging Technologies in IEEE International Conference on Engineering Technologies, Volume, Issue, 13-14.

[10] M C Chan, J R Jiang and S T Huang, "Fault Tolerance and Secure Network storage", 7th International Conference on Digital Information Management, ICDIM, (2012).

[11] Gang chen, Hai Jin, Deqing Zou, Bing Bing Zhou, Weizhong Qiang, Gang Hu, SHelp:Automatic Self Healing for Multiple Application Instances in a Virtual Machine Environment", IEEE International Conference on Cluster Computing 2010.

[12] Imad M Abbadi, "Self Managed conceptual Model in Trustworthy Clouds Infrastructure", 2010.

[13] Yang Zhang1, Anirban Mandal2, Charles Koelbel1 and Keith Cooper, "Combined Fault Tolerance and Scheduling Techniques for Workflow Applications on Computational Grids", 9th IEEE/ACM International Symposium On Clustering and Grid, 2010.

[14] Michael Armbrust, Armando Fox, Rean Griffith, "Above The Clouds:A Berkeley View of Cloud Computing", Electrical Engineering and Computer Science University of California at Berkeley, 2009.

Received: 15 May 2019/Accepted 05 July 2019.

[15] Wenbing Zaho, P. M. Melliar-Smith and L. E. Moser" , Fault Tolerance Middleware for Cloud Computing", 2010, IEEE 3rd International Conference on Cloud Computing.

[16] Talwana Jonathana Charity And G C Hua, Resource Reliability Using Fault Tolerance In Cloud Computing, 2nd International Conference On Next Generation Computing Technologies (Ngct) 2016.

[17] Bashir Mohammed, Mariam Kiran, Iffan Ullah Awan, K. M. Maiyama, An Integrated Virtualized Strategy for Fault Tolerance In Cloud Computing Environment, International. IEEE Conferences on Ubiquitous Intelligence and computing, Advanced and Trusted Computing, Scalable computing and Communications, Cloud and Big Data Computing, Internet of People and Smart word Congress (UIC/ATC/SCalCom/CbdCom/IoP/SmartWorld) 2016.

[18] Emam AbdElfattah, Mohamed Elkawkagy, and Ashraf, El-Sisi, A Reactive Fault Tolerance Approach for cloud computing 13th International Computer Engineering Conference (ICENCO) 2017.

[19] Samir Setaouti, Dijamel Amar Bansaber, Reda Adjoudj, and Mohamed Rebbah, Fault Tolerance Model Based On Service Delivery Quality Levels In Cloud Computing, International Conference On Mathematics Information Technology (ICMIT) 2017.

[20] Umesh Dwivedi, Harsh Dev, A Review On Fault Tolerance Techniques and Algorithms in Green Cloud Computing, journal Of Computational and Theoretical Nenoscience, Vol 15, 1-12, 2018.

Received: 15 May 2019/Accepted 05 July 2019.