

# Performance Analysis of Reliable VM Identification Using Resource Availability Method for Cloud Computing

N.Chandrakala<sup>1</sup>, Dr.P.Sivaprakasam<sup>2</sup>

<sup>1</sup>Ph.D Research Scholar, Mother Theresa Women's University

<sup>2</sup>Sri Vasavi College, India

nchandrakala15@gmail.com psperode@yahoo.com

**Abstract** - Cloud computing is one of the latest emerging trends in Information technology. Cloud computing shares the resources through the internet. Today there is a demand for cloud computing because of this resource sharing. This demand causes many challenges in cloud computing research areas. One of the major aspects of the cloud computing challenges is fault tolerance. The fault tolerance can be achieved in many ways. Our proposed method identifies the fault based on the resource availability of Virtual machines (VM). The reliability of the VM can be identified based on the response time or resource availability. If the VM responds with in the time that VM is taken as reliable. If the time limit exceeds the previous history will be taken. Suppose if two VM has respond same time limit the reliability can be identified based on the resource availability. Sometimes the data is received by the VM is corrupted in this situation the reliable time of the VM is taken but it is not considered for reliability assessment. The resource availability is the one of the proposed techniques used in this paper.

**Keyword** - Cloud computing, Fault tolerance, Reliable, Decision mechanism, Resource availability.

## I. Introduction

The "Cloud" is the default symbol in internet diagram. The term "Computing" refers to Computation and storage. Cloud computing shares the resource like physical services, storage, and networking. The cloud offers many services through cloud service providers. The most popular cloud service providers are Google, Amazon, Salesforce etc...Each service provider provides different services based on the demand of the users. For example Amazon provide IaaS service Google Provides all services like SaaS, PaaS and IaaS.

The cloud computing is based on the distributed concepts and it is reliable to all users. This paper deals the research field of fault tolerance. The fault can identify speed and available memory size. In a cloud environment there are many unknown nodes called Virtual machines(VM). Virtual machine (VM) is an operating system (OS) or program can be installed and run virtually. Other term VM is a processing

machine in the server. In cloud computing the user data is replicated in many VM's.

The client request is passed all the available VM's. If the particular VM is failure that the VM is not responding all the other active VM respond the request. The cloud computing can identify the one reliable among all the VM's and respond the client request. This paper is used to identify the reliable VM.

## II. Related Work

Lot of research work was done in fault tolerance area. The resource availability technique is not used in the previous research. Adaptive Fault Tolerance in Real Time Cloud Computing AFTRC by Sheheryar Malik abd Fabrice Huet. This paper deals the reliable VM identification using time taken method and based on IP address. The IP address is taken if the two VM has given same response time. No resource availability method used in this paper. The proposed method deals the reliability of the VM is identified based on the time taken and the resource availability techniques. The fault tolerance in this paper is done on reliability of virtual machines response in the cloud environment[1][2][3]

## III. PROPOSED MODEL

The proposed method is Performance analysis of Reliable VM identification in Resource availability techniques for cloud computing. The reliable VM is identified based on the time, previous history and the resource availability. The proposed method uses fuzzy set logic to create the private ID for each client request in a cloud environment. The use of private ID is to avoid the conflict of same file name stored in the cloud environment. The propose method use two parts. One is the set of virtual machines and other is adjudication node such as the main server. The Virtual machine uses acceptance test algorithm for its logical validity. The adjudicator contains the time checker, reliability assessor and decision mechanism algorithms to find the reliable VM. After process, the reliable VM is identify to process the client request. The client can accept the data from the VM is in compressed form. A virtual machine is selected for computation on basis of reliability and can be removed, if does not perform well.

The output of a node with highest reliability is selected as the system cycle output. The data owner owns data from the client and store in cloud. To avoid the fault tolerance the data can d store it on multiple cloud using indexing mechanism. The requested user is valid the data owner generates the private id and sends it. If the user is not the valid data owner will drop the request message[4][5]

**Client:** If any users want this data send a user authentication message to the data owner. In Client, after receiving private id, it sends the data requisition message to Multiple VM's (Virtual Machine), CSP (Cloud Service Provider).

**CSP:** In CSP, they can check the requested user private id and data, if it's valid send that requested data to acceptance

**Acceptance test (AT):** Check whether the fault can occur or not. Here the fault is a failure of VM or files stored in corrupted form. The acceptance test can respond both success and failure case of the VM. If the VM has failed that VM is not considered. Suppose the VM is not failure but the data is corrupted means that VM is considering for time checker.

**Reliable Communication(RC):** The RC Identify the reliable VM among the VM's in cloud environment . The reliable VM is identified by the time Checker mechanism. Time checker module checks

1. The accessing time
2. Memory Availability

**Reliability Assessor(RA):** The RA module assesses the reliability for each virtual machine. The reliability is identified based on the main core module of the proposed system. As the proposed system tolerates the faults and makes the decision on the basis of the reliability of the processing nodes (i.e. virtual machine).

The reliability of the virtual machine is adaptive, which changes after every computing cycle. In the beginning the reliability of each virtual machine is 100%. If a processing node manages to produce a correct result within the time limit, its reliability increases. And if the processing node fails to produce the correct result or result within the time, its reliability decreases. The reliability assessment algorithm is more convergent towards failure conditions. In RA, the VM response above the time limit that VM is considering as false identification.

**Decision Mechanism:** Identify the Reliable VM's based on

- Time consuming (Milliseconds)
- Resource availability.
- Previous history.

**Time consuming:** Time consuming is calculated in milliseconds. The time limit is given as a lower bound time limit and the upper bound time limit. In our process the response time limit for each VM is given milliseconds. The VM can respond within the specified time limit that VM is taken as reliable.

**Memory Availability:** Memory is taking as resource. The memory availability for each VM is taking separately. The two or more VM has respond at a time. We apply the memory availability algorithm to find the lowest memory utilization VM and take that VM as reliable. The resource availability was taken at the time of VM has respond the same time. In this situation we take the resource of two VM and identify the high memory available VM as reliable.

**Previous History:** All the previous reliable VM details are stored in the server .database. At the time of processing all VM respond exceed time limit we go for previous history. The reliable VM is taking as the number of time particular VM is reliable in the previous processes is as taken as a count. For example theVM1 has store more times in the database than the other VM's then VM1 is taken as reliable. The figure 1.Shows the system model of the proposed method.

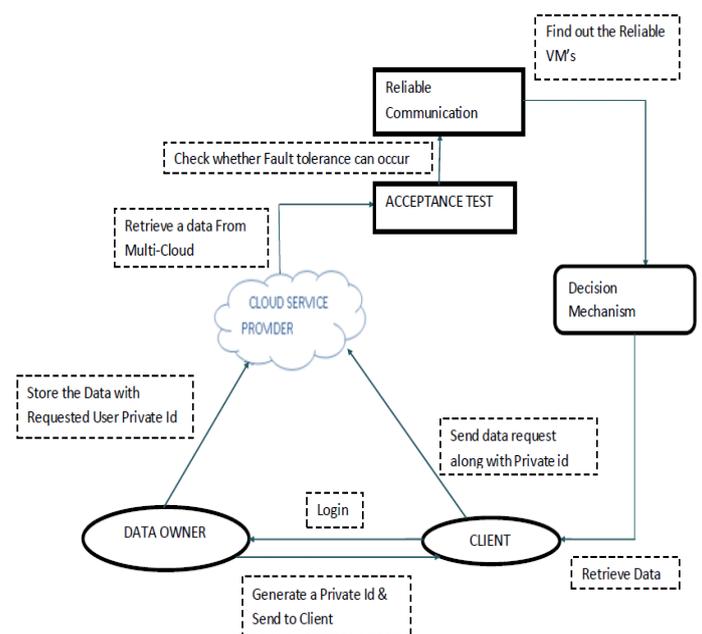


Figure 1. System Model

#### IV. EXPERIMENTS AND RESULTS

We take four VM's on this experiment. The virtual machines VM1, VM2, VM3 and VM4 are configured. Each virtual machine runs a diverse algorithm such as algorithm1, algorithm2 algorithm3 and algorithm4. The adjudication node sends data to VM's and receiving the results from the VM's. The response time for the VM is taken as lower limit of 1 to upper limit of 1.5 milliseconds. All VM executes the algorithm simultaneously. The VM1 checks the data availability and retrieval for testing the validity of user and the acceptance test is used to check the availability of the data. The same process is done in VM2, VM3 and VM4. After processing this mechanism the result is passed to time checker module. In this module the time consumption for each VM is retrieving and pass the result to the reliability assessor module. The reliability assessor identifies the Reliable VM and pass the result to the decision mechanism module. Decision Mechanism has taken the reliable VM based on the conditions given in the algorithm. The decision mechanism can take the reliable VM based on the memory availability. Another method used in our proposed method is fuzzy logic[6][7]. This method is used at the time of data stored in the VM each file is stored in the cloud environment has one private ID. The purpose of this ID is to avoid conflict of same name used by different users in a public cloud environment. Only the owner of the data can know the private ID. The client uses the private ID instead of file name. This private ID is not predicted. The data send the client and VM is in compressed form. The security is also high in this method[8][9][10].

$$(MEM)_{avi} = \Sigma(TOT)_{mem} - (MEM)_{U}$$

$$\Sigma(TOT)_{mem} \rightarrow \text{Bytes}$$

$$(MEM)_{avi} \rightarrow \text{Bytes}/1024$$

$$\Sigma i <= 4 \rightarrow \Sigma(MEM)_{avi} (VM1, VM2, VM3, VM4)$$

This parameter describes unbalancing factor for machine p considering acceptance of request i.

Equation for finding Reliability:

$$P_{reqt} = \text{get current time};$$

$$P_{del} = \text{get endtime};$$

$$T_{pt} = T_{Reqt} + T_{Delivered}$$

$$T(pt) = \sum_{n=4}^l (P_{reqt} + P_{del})$$

$$T_{pt} = \sum p_{t1} + (T_{Reqt} + T_{Delivered}) + \sum p_{t2} + (T_{Reqt} + T_{Delivered}) + \sum p_{t3} + (T_{Reqt} + T_{Delivered})$$

$$R = \sum_{n=4}^{Rt \leq 1.5} T_{pt} \leq 1.5$$

Where,  $P_{reqt}$  = get the current start time,  $P_{del}$  = get the current end time.  $T_{(pt)}$  = total processing time.

$R$  = reliability.  $N$  = total number of available virtual machine

Where,  $P_{reqt}$  = get the current start time,  $P_{del}$  = get the current end time.  $T_{(pt)}$  = total processing time.

$R$  = reliability.  $N$  = total number of available virtual machine

**Begin**

**Input Threshold**

$Best_{R_i}$  = **find\_reliability** of node with highest reliability among all VM's.

**if**  $Best_{R_i} > \text{Threshold}$

Data: = loss

**Else**

Data: = Gives data to Requested user

**End**

This algorithm can be used to find out the best reliability time duration between among all VM's, based on the result corresponding data can be delivered to the perspective user from highly Reliable VM. Where the threshold denotes some predefined criteria,  $Best_{R_i}$  denotes the best reliable value among various VM's.

**Begin**

$a := 1, \text{Status}, \text{Vm} := 4;$

**Input**  $N_{pt}$

**While**  $vm \geq 4$

$T_{pt} := T_{Reqt} + T_{Delivered}$

$R_i := T_{pt}$

**If**  $R_i < N_{pt}$  **then**

$a := a - 1$

Status = Active

**Else if**  $R_i > N_{pt}$  **then**

$a := a + 1;$

Status := dead

**Wend**

**End**

#### V. EVALUATION

The Evaluation of the four Virtual machine is taken at the time and the memory utilization. The time is taken in milliseconds. Here the VM4 is the least memory utilization in this experiment. The VM1 is not considered because it gives the false result in the reliability assessment test. The VM2 is not taken as reliable because it takes more time than the limit. For the final evaluation the VM3 and the VM4 is taken for decision mechanism because the VM3 and VM4 respond within the time but these two gives the same time limit. So we take minimum memory consumption VM. finally VM4 is taken as reliable In this process the success node is taken to identify the processing time. The processing

speed is increased when the VM has less load. The processing speed is calculated as below. The processing speed is denoted as ps. The time is denoted as t and the memory load is taken as ml. The processing time is calculated as  $\tau = ml/ps$

The processing time is taken only for the success request to the VM. The request is sent first time  $pt=1$ . At the time of sending request when the VM is a failure the request send again until the success of the VM.  $Pt = 1,2,3,\dots,N$ . We calculate this in processing time as

$$T_p = T_1 + T_2 + \dots + T_N + \tau$$



Figure 2: Memory Consumption Chart

The figure 2 shows the performance analysis of the VM. Here VM4 contains more memory space, in this situation the VM4 response quickly so the here the VM4 is taken as reliable.

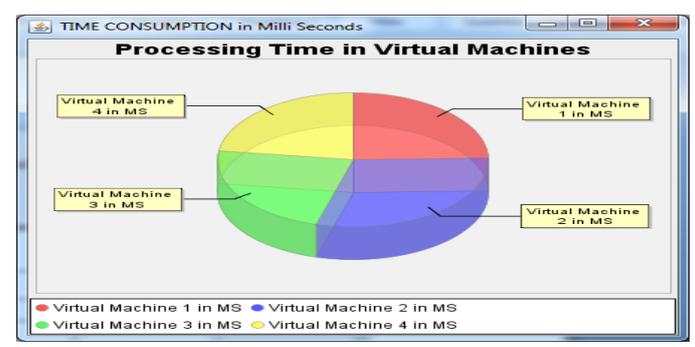


Figure 3: Time taken Chart

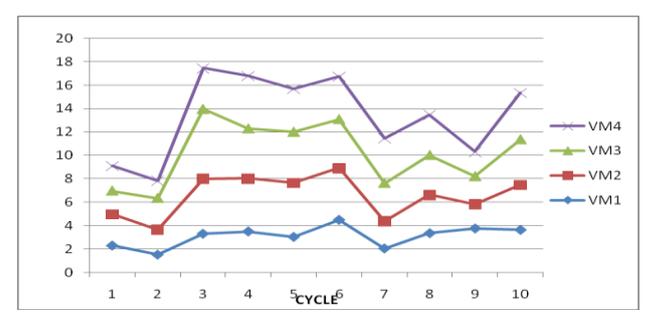


Figure 4: Speed Analysis

The Figure 4 Shows the evaluation report of the VM's in 10 computation cycle based on the memory consumption and the time taken. The evaluation result shows the most of the VM give high performance when we take memory utilization method. On time taken method most of the VM does not responded within the time or it respond same time. Finally the result shows the high memory consumption VM gives the better performance.

CYCLES	VM1			VM2			VM3			VM4			RELIABLE
	TIME	MEMORY	SPEED	TIME	MEMORY	SPEED	TIME	MEMORY	SPEED	TIME	MEMORY	SPEED	
1	1.531669	223978	2.329521	1.140116	240068	2.655626	1.923222	2078881	2.003413	1.84404	2178180	2.103413	VM2
2	1.710712	193768	1.556773	1.477385	209356	2.098734	1.230523	2177952	2.701122	1.88059	1668681	1.463596	VM3
3	1.637992	289732	3.326561	1.695849	508784	4.684829	1.148599	523780	5.947768	1.84779	4477880	3.483932	VM3
4	1.548482	340538	3.524945	1.91625	447680	4.492482	1.649393	495004	4.276508	1.77248	4532845	4.496301	VM3
5	1.429497	239040	3.063322	1.213989	467844	4.599928	1.562091	444272	4.347866	1.66383	4021480	3.667146	VM2
6	1.012958	452876	4.507524	1.119811	446944	4.422666	1.904673	423616	4.155725	1.82033	3949001	3.657118	VM1
7	1.782822	371342	2.084016	1.743388	424752	2.305726	1.475557	491096	3.249397	1.11561	4990567	3.820904	VM4
8	1.497138	390704	3.374611	1.456761	354236	3.261159	1.640205	326840	3.388726	1.27898	3098002	3.437812	VM4
9	1.009473	437924	3.777563	1.235779	252624	2.064391	1.268995	263620	2.379296	1.09729	4218512	2.082021	VM1
10	1.542076	309124	3.835233	1.428183	343874	3.874054	1.380557	419996	3.974583	1.63855	3745688	3.659138	VM3

Table 1: The Reliable VM Identification in computation Cycle

The table 1 shows the computation cycle of the VM's and identify the reliable VM among the four in each computation cycle. Here in this computation we take only the

true condition satisfied VM in reliability assessment procedure. Each and every cycle the reliability of the VM is

changed. The memory utilization of the reliable VM also shown in the table.

## VI. CONCLUSION AND FUTURE WORK

The replica of all original files is stored it on VMs. If suppose one VM can suffered by fault, others can produce the result efficiently Based on a reliability assessment algorithm and decision mechanism algorithm. The proposed method gives the high reliability of the Virtual machines Maximum the reliability is taken as based on the resource of the VM. The time complexity of the process is also minimized because it will not use any backward method. Once the resource checker is used the reliability is taken based on the result. The time limit for the VM is allotted as less so that the response of the VM is taken as immediately this is one of the efficiency concepts used in this method. It has a dynamic behavior of reliability configuration. The reliable VM identification technique used in this process is very efficient to improve the QOS of cloud . In future the scheduling is used to identify the reliability.

## VII. REFERENCES

- i. X. Kong, J. Huang, C. Lin, "Comprehensive Analysis of Performance, Fault-tolerance and Scalability in Grid Resource Management System" - Eighth International Conference on Grid and Cooperative Computing, Lanzhou, China, August 27-29, 2009
- ii. B. Yang, F. Tan, Y. S. Dai, and S. Guo, "Performance evaluation of cloud service considering fault recovery," Proc. of the First International Conference on Cloud Computing, Beijing China, December 2009, pp.571-576.
- iii. Jing Deng Scott C.-H. Huang, Yunghsiang S. Han and Julia H. Deng, "Fault-Tolerant and Reliable Computation in Cloud Computing"- IEEE Globecom Workshop on Web and Pervasive Security 2010
- iv. Jibe Li, Marty Humphrey, You-Wei Cheah and Youngryel Ryu, "Fault Tolerance and Scaling in e-Science Cloud Applications: Observations from the Continuing Development of MODISAzure" - IEEE e-Science 2010
- v. Zibin Zheng, Tom Chao Zhou, Michael R. Lyu, and Irwin King, " FTCloud: A Component Ranking Framework for Fault-Tolerant Cloud Applications" - IEEE 21st International Symposium on Software Reliability Engineering 2010.
- vi. Y.M. Teo, B.L. Luong, Y. Song<sup>2</sup> and T. Nam, "Cost-Performance of Fault Tolerance in Cloud Computing" - Special Issue of Journal of Science and Technology, Vol. 49(4A) October 2011.
- vii. Sheheryar Malik and Fabrice Huet , " Adaptive Fault Tolerance in Real Time Cloud Computing" - IEEE World Congress on Services 2011
- i. Vahid Khoshdel and Seyed Ahmad Motamedi, Saeed Sharifian, "A New Approach for Optimum Resource Availability in Cloud Computing Environments"- International Conference on Computer and Knowledge Engineering Oct. 13-14, 2011
- ii. JiSu Park, HeonChang Yu, KwangSik Chung and EunYoung Lee, " Markov Chain based Monitoring Service for Fault Tolerance in Mobile Cloud Computing" - Workshops of International Conference on Advanced Information Networking and Applications 2011.
- iii. Yilei Zhang, Zibin Zheng and Michael R. Lyu, "BFTCloud: A Byzantine Fault Tolerance Framework for Voluntary-Resource Cloud Computing" - IEEE 4th International Conference on Cloud Computing 2011
- iv. Yang, F. Tan, and Y. S. Dai, "Performance evaluation of service considering fault recovery," The Journal of computing Supercomputing, pp. 1-19, 2011.
- v. Anju Balal and Inderveer Chana<sup>2</sup>, "Fault Tolerance-Challenges, Techniques and Implementation in Cloud Computing" - IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012.
- vi. E. Feller, L. Rilling, C. Morin, "Snooze: A Scalable and Autonomic Virtual Machine Management Framework for Private Clouds", December 2011.
- vii. Y. M. Teo, B. L. Luong, Y. Song, T. Nam, "Cost-Performance of Fault Tolerance in Cloud Computing", October 28, 2011