ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Resource Reliability using Fault Tolerance aware Scheduling in Cloud

Diptee H. Devmurari[1] , Prof. Kashyap Raiyani[2]

M.E. Student, Dept. of Computer Engineering, Marwadi Education Foundation's Group of Institutions, Rajkot, Gujarat, India[1]

Professor, Dept. of Computer Engineering, Marwadi Education Foundation's Group of Institutions, Rajkot, Gujarat, India [2]

ABSTRACT: Cloud computing offers variety of services from software instance to resource provisioning. As the user demands increase, there is a necessity to enhance cloud offerings. But still in some cases fault-tolerance is the major challenge for cloud environment. Multiple request to access the same server sometime leads to server over loaded and may increase faults and cause unreliability for the server. In this paper, we propose pro-active approach for fault-tolerance based on Processing power, Memory and Network parameters to increase resource reliability. Through this approach, we first calculate the reliability of each Virtual Machine (VM) based on success rate of task execution and then schedule the task on highly reliable VM. This approach provides comparatively good results for the VM reliability and system stability.

KEYWORDS: Fault Tolerance, Reliability, Service Level Agreement, Quality of Service

I. INTRODUCTION

Now a day, because of demand and benefits of the cloud computing infrastructure increased, cloud infrastructure is mainly used for real time computing. In the real time applications of cloud, most of the processing is executed on remote computing nodes. So in this situation, applications which are running on cloud, needs abilities to tolerant the faults so that it can reduce the chances of system failure so that system can work properly their functions when failure occurs.

The correctness of the applications which are running on the system not only depends on the logical result, it also depends on the time it was delivered[3]. If system fails to give response, it is the bad as the wrong response[4]. There are two basic characteristics of the system one is fault tolerance and other is timeliness. Fault tolerance means that system should continue to operate even the fault is present.

However, these environments are prone to performance variations and different types of failures (e.g., in resources or in platforms). Failures also affect the overall workflow execution time by increasing its make span. Failures in a workflow application are mainly of the following types: task failures, machine (VM) failures and hardware failure. Task failures may occur due to dynamic execution environment configurations, missing input data, or system errors. Machine (VM) failures are caused by hardware failures and load in a distributed system, among other reasons, hardware failure are caused by power failure or any physical damage.

For this purpose, we propose a model for the fault tolerance of systems running at cloud infrastructure. This model will be mainly determining timeliness of the output and reliability of the systems. Our proposed method identifies the fault based on the resource availability of Virtual Machines (VM). The reliable VM is identified based on the time, previous history of that VM and the resource (CPU, Memory and available bandwidth) availability.

We have proposed an approach for resource reliability using fault tolerance that is based on pro-active technique fault-tolerance without denial of resources to the clients. Reliability of VM is calculated based on Memory, CPU, and bandwidth and then new task will be submitted to reliable VMs. We also proposed an algorithm based on ratio mathematical equations by containing all areas like Memory (RAM), CPU (MIPS) and Network together. In our

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

approach task will be submitted to most reliable running VM and if reliability is not available then it will create new VM and submit task to that new VM.

II. Related work

Deepak. P.C has provided Robust and Fault-Tolerant Scheduling approach for Scientific Workflows in Cloud Computing Environments [6].In his approach, there is a fix time limit given to each tasks. If the task does not execute within specified deadline, then it would be discarded to lead performance by increasing execution speed. But the drawback of this approach is Quality of Service (QoS). This approach definitely increase performance but will compromise with QoS issues as few tasks which are not executed within deadline will be discarded. Hence, this approach does not provide guarantee to execute 100% execution of allocated tasks on available resources [7].

During execution process, to maintain budget by minimizing the cost for execution [8], we have to first identify all available requests / tasks and their respective budget to execute them by calculating time and resource requirements. Then, it will arrange the tasks in ascending order based on their required cost. After this, it will start execution in arranged order. It increases overhead as replica always needs storage and execution of tasks / requests at all locations or servers basically. So, it increases all the required resources like CPU (MIPS), Memory, Bandwidth and Server for storage. It always increases the execution time. During execution of larger job, if execution engine fails to execute the task as 100% then this approach prefers to restart the execution from the beginning. So the executed percentage of job in previous scenario will be executed again.

A Dynamic Data Fault-tolerance Mechanism for Cloud Storage [9, 10, 11], Liying Wu et.al have used the approach to apply check points and message logging. So, while sending any larger message towards server, it always creates check points at certain intervals hence it needs high accuracy and continuous monitoring. To keep message logging by applying check points, it consumes storage, memory and CPU resources and decrease the performance of the available resources. Review on fault tolerance techniques in cloud computing [7, 12] studied different parameters based on following techniques:

A. Proactive Fault tolerance

The Proactive fault tolerance policy is to avoid recovery from fault, errors and failure by predicting them and proactively replace the suspected component means detect the problem before it actually come.

- Check point / restart
- Replication
- Job migration
- Task resubmission
- Exception handling

B. Reactive Fault tolerance

Reactive fault tolerance policies reduce the effort of failures when the failure effectively occurs. This technique provides robustness to a system.

- Self-healing
- Preemptive migration

Anju Bala et al. has provided the idea of designing an intelligent task failure detection models [13, 14] for facilitating proactive fault tolerance by predicting task failures approach. The working of model is distributed in two modules. In first module, task failures are predicted with machine learning approaches and in second module, the actual failures are located after executing workflow execution in cloud test-bed. Machine learning approaches such as Naïve Bayes, ANN, logistic regression and random forest are implemented to predict the task failures intelligently from the dataset of scientific workflows.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Anjali Meshram et al. provided fault tolerance model for cloud (FTMC) approach [15]. This model accesses the reliability of computing nodes and chooses the node for the computation on the basis of reliability. The node can be removed if it does not perform well.

ShivamNagpal et al. proposed a fault tolerant model that takes decisions [16]. Reliability of a node is estimated on the basis of 2 parameters; accuracy and time. If any of the nodes does not achieve the level, then backward recovery is performed by the system. This model focuses on adaptive behavior of processing nodes and the nodes are removed or added on the basis of reliability.

III. PROPOSED ALGORITHM

The proposed system model designed to find out the reliability of each VM and then cloudlets will be assigned to best reliable VM. Initially, Fig. 1, client sends requests / tasks to cloud broker. The task of the cloud broker is to assign cloudlets to Cloud Provider.

To check the reliability of VM, first check whether cloudlets are successfully complete their execution or failed within time deadline. Based on Success and Failure cloudlet, update reliability of each VM at the end, select the best reliability Virtual Machine from existing VMs and assign cloudlets on it. Therefore in our system model, we repeat all the steps for each cloudlet.

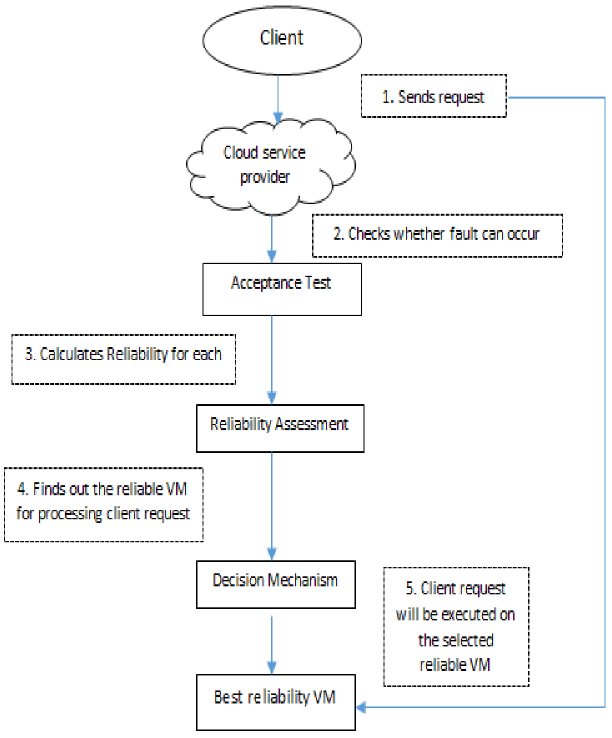
Client
1. Sends request
Cloud service provider
2. Checks whether fault can occur
Acceptance Test
3. Calculates Reliability for each
Reliability Assessment
4. Finds out the reliable VM for processing client request
Decision Mechanism
5. Client request will be executed on the selected reliable VM
Best reliability VM

Fig1. System Model Flow Diagram

IV. PSEUDO CODE

Denotation

MM_i	Available Memory ratio of ith VM
CP_i	Available CPU ratio of ith VM
BW_i	Available Bandwidth ratio of ith VM
Available RAM_i	Available RAM in ith VM
Available $MIPS_i$	Available MIPS in ith VM
Available BW_i	Available Bandwidth in ith VM
Total RAM_i	Total RAM in ith VM
Total $MIPS_i$	Total MIPS in ith VM
Total BW_i	Total Bandwidth in ith VM

$$\text{Reliability of VM} = \frac{\sum_{\text{cluodleti}} RAM_i + CPU_i + BW_i}{\text{No.of Cloudlet}} \quad (1)$$

$$RAM_i = \frac{\text{AvailableRAM}_i}{\text{Total}_{RAMi}} \quad (2)$$

$$CP_i = \frac{\text{AvailableMIPS}_i}{\text{Total}_{MIPSi}} \quad (3)$$

$$BW_i = \frac{\text{Available } BW_i}{\text{Total}_{BWi}} \quad (4)$$

$$R_i = \frac{CPU_i + MM_i + BW_i}{3} \quad (5)$$

$$\text{Reliability of host} = \frac{\sum_{vmi} \text{reliability}_{vmi}}{3} \quad (6)$$

A. Reliability Calculation Algorithm:

Ri= Reliability of ith VM
Input: cloudlets, VM
Output: Reliability of VM and Reliability of host

```
--------------------------------------------------------------------------------------------------------------
Begin
Input Number of VMs, Number of Cloudlets
Foreach Cloudlet
Foreach VMi
CPi= Available MIPS / Total MIPS
MMi=Available RAM / Total RAM
BWi= Available BW / Total BW
Ri = ( CPi + MMi + BWi) / 3
End for
Sort VM reliability in decreasing order and store it in VMr
Foreach vm in VMr
If (Allocate cloudlet in vm == success)
          Allocate cloudlet on vm
Break
End if
End for
End for
```

V. SIMULATION RESULTS

Here we will analyze the results of proposed approach. Then we will compare the results of proposed mechanism with the existing approach. In our simulation environment, we have submitted various numbers of cloudlets. We have submitted 50, 150, 200, 250, 300 numbers of cloudlets. In result we get number of failed cloudlets. Table 5.2.1 shows comparison of failed cloudlet in existing method and proposed method.

	Failed cloudlet	
No. of Cloudlet	Existing method	Proposed method
50	2	0
100	5	0.4
150	6	3.2
200	8	5.8
250	12	10.7
300	15	13.3

Table 5.2.1 Comparison of failed cloudlet in existing method and proposed method

Here from above table we can see that there is less number of failed cloudlets using proposed method in compare to existing method. So we can say that proposed method is better than existing one. So based on proposed algorithm, scheduler will find most reliable VM and new incoming task will submitted on most reliable VM. Hence, proposed method will improve fault tolerance compare to the existing method.

Fig.5.2.1 shows the graph for comparison of failed cloudlet in existing method and proposed method.

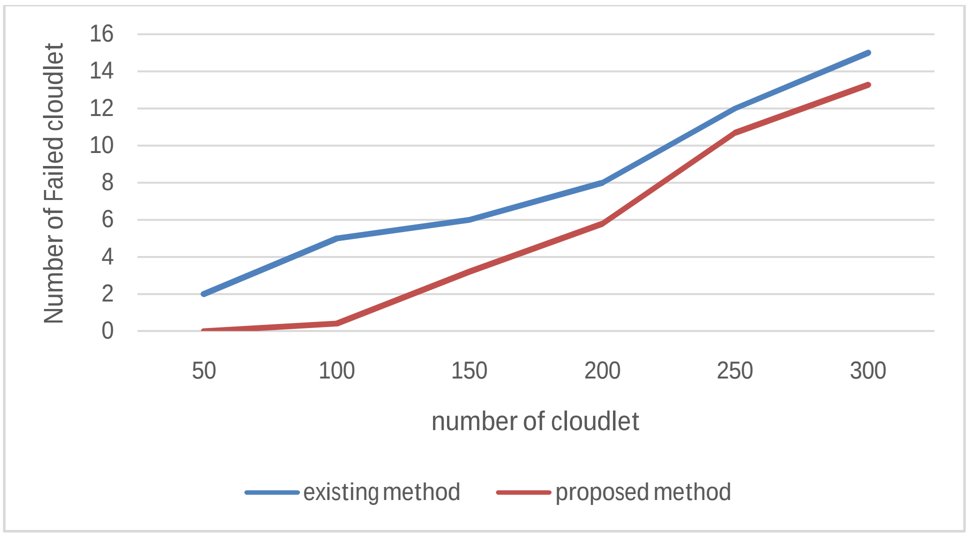

Fig.5.2.1 Comparison of failed cloudlet in existing method and proposed method

Fig.5.2.2 shows the comparison of average failed cloudlet in existing method and proposed method.

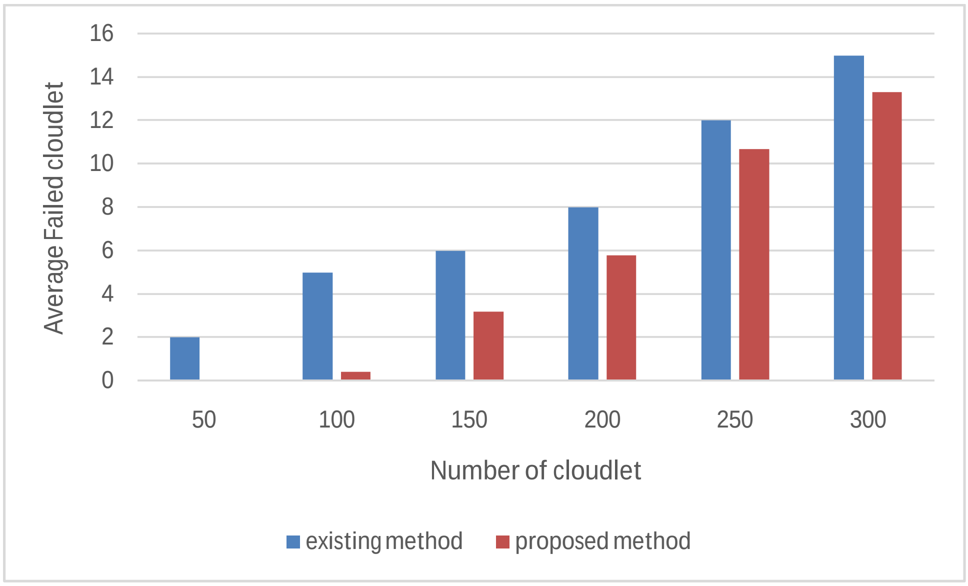

Fig.5.2.2 Comparison of average failed cloudlet in existing method and proposed method

VI. Conclusion and Future Work

In cloud Data center, multiple request to access the same server sometime leads to server overloaded and may increase faults and cause unreliability for the server. In our proposed work we enhance resource reliability using fault tolerance for cloud environment on the basis of three parameters processing capacity, memory and bandwidth and based on this algorithm scheduler will find most reliable VM and new incoming task will submitted on most reliable VM. It is demonstrated that failed cloudlet in proposed method is significantly reduce compare to existing method. Here we consider three parameters for enhancing resource reliability: processing capacity, memory and bandwidth. We can add more parameters for checking the reliability of VM.

References

1. Anju Bala and Inderveer Chana. “Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing”. IJCSI International Journal of Computer Science, 9(1), January 2012.
2. Inc. Sun Microsystems. “Introduction to Cloud Computing Architecture”.
3. P. Latchoumy and P. Sheik AbdulKhader. “Survey on Fault Tolerance in Grid Computing”. IJCSI International Journal of Computer Science, 2(4), November 2011.
4. Wenbing Zhao, P.M. Melliar, and L.E. Mose. “Fault Tolerance Middleware for Cloud Computing”. In Proceedings of IEEE 3rd International Conference on Cloud Computing, 2010.
5. Dutch Meyer, Mike Feeley, Norm Hutchinson, Brendan Cully, Geoffrey Lefebvre and Andrew Warfield. “REMUS: High Availability via Asynchronous Virtual Machine Replication”. In Proceedings of 5th USENIX Symposium on Networked Systems Design USENIX Association and Implementation, 2008.
6. Deepak PoolaChandrashekar. “Robust and Fault-Tolerant Scheduling for Scientific Workflows in Cloud Computing Environments”. In 28th International Conference on Advanced Information Networking and Applications, August 2014.
7. Zeeshan Amin, Nisha Shethi, Harshpreet Singh. “Review on Fault Tolerance Techniques in Cloud Computing”. International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 18, April 2015.
8. G. Juve, A. Chervenak, E. Deelman, S. Bharathi, Gaurang Mehta, and Karan Vahi. “Characterizing and Profiling Scientific Workflows”. Future Generation Computer Systems, 29(3):682 – 692, 2013.
9. Liying Wu, Bo Liu, Weiwei Lin. “A Dynamic Data Fault-tolerance Mechanism for Cloud Storage”. 2013.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

10. S.Abrishami,M.NaghibzadehandD.H.J.Epema. "Cost-Driven Scheduling of Grid Workflows Using Partial Critical Paths". IEEE Transactions on Parallel and Distributed Systems, 23(8):1400 –1414, August 2012.
11. S. Abrishami, M. Naghibzadeh, and D.H.J. Epema. "Deadline Constrained Workflow Scheduling Algorithms for Infrastructure As a Service Clouds". Future Generation Computer Systems, 29(1):158 – 169, 2013.
12. D.Kondo,B.Javadi,A.IosupandD.Epema. "The Failure Trace Archive: Enabling Comparative Analysisof Failuresin Diverse Distributed Systems". In10thIEEE/ACMInternationalConference on Cluster, Cloud and Grid Computing (CCGrid), 2010, pages 398–407. IEEE, 2010.
13. Kaushal, V. and Anju Bala"."Autonomic Fault Tolerance Using Haproxy in Cloud Environment". International Journal of Advanced Engineering Sciences and Technologies, 7(2), 54-59, 2011.
14. Anju Bala andInderveer Chana. "Fault Tolerance Challenges, Techniques and Implementation in Cloud Computing". International Journal of Computer Science Issues (IJCSI), 9(1), 2012.
15. Anjali. D. Meshram, A. S. Sambare, S. D. Zade."Fault Tolerance Model in Cloud Computing". IJAIS Proceedings on 2nd National Conference on Innovative Paradigms in Engineering and Technology (NCIPET 2013), 2013.
16. ShivamNagpal, Praveen Kumar. "A Study on Adaptive Fault Tolerance in Real Time Cloud Computing". International Journal of Advanced Research in Computer Science and Software Engineering, 2013.
17. Saurabh Kumar Garg, Adel NadjaranToosib, Srinivasa K. Gopalaiyengarc, RajkumarBuyyab. "SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter". International Journal of Network and Computer Application, ELSEVIER 45, 2014.