International Journal of Advance Engineering and Research Development (IJAERD)
Volume 1,Issue 5,May 2014, e-ISSN: 2348 - 4470 , print-ISSN:2348-6406

Weighted RF Based Fault Tolerance for Smarter Planet-Cloud Computing

Sweta Patel[1], Prof. Hitesh Bheda[2]

[1]Dept. of Computer Engineering, RK University, Rajkot, India Sweta.patel27@gmail.com
[2]Dept. of Computer Engineering, RK University, Rajkot, India

***Abstract*—**Cloud Computing aims to provide reliable services within data centres that contain servers, storage and networks. The services are delivered to the users transparently without their need to know the details of the underlying software and hardware. One of the challenge of cloud computing is to ensure that the applications run without a hiatus in the services they provide to the users. In most of the real time cloud applications, processing is done on remote cloud computing nodes. So there are more chances of errors. Also, the correctness of the applications running on the system depends not only on the logical result, but also the time it was delivered. Failure to respond is as bad as the wrong response. These systems have two basic characteristics that is Timeliness & Fault tolerance. For this purpose, we have presented a model for the fault tolerance of systems that mainly determines 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 reliability of the VM can be identified based on the response time or resource availability. The resource availability is the one of the proposed techniques used in this paper.

Keywords— Fault tolerance as a Service, System level Fault tolerance, Cloud Computing, Fault tolerance factors, Cloud Reliability, VM reliability, Fault Independence

I. Introduction

Cloud computing can be defined as a way of using computational resources such as storages, operating systems etc. which are located remotely and are provided as a service over internet [1]. The elementary advantages of cloud computing includes low costs, high availability, scalability and elasticity.

On the other hand, the reliability of Cloud computing still remains a major concern among users. Due to economic pressures, these computing infrastructures often use commodity components exposing the hardware to scale and conditions for which it was not originally designed [2]. As a result, significantly large number of failures manifest in the system and seemingly impose high implications on the hosted applications, effecting their performance and availability. In this context, applications require fault tolerance abilities so that they can overcome the impact of system failures and perform their functions correctly when failures happen [3].

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 the reliability of the virtual machines based on the weightage of the available memory size and the processing capacity. 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.

II. related work

The use of cloud infrastructure for real time computing is quite new. Most of the real time applications require the fault tolerance capability to be provided. A lot of work has been done in the area of fault tolerance for real time application. But there is lot of research scope available in fault tolerance for real time application running on cloud infrastructure. Cloud infrastructure has introduced some new issues related to real time computing. The user of real time cloud applications has lose control over the nodes. He does not know where his application is going to be processed [4]. But on the brighter side, cloud has a facility to scale up dynamically. So the faulty node (node represents itself and the communication link) can be removed. A new node can be added, if required. These characteristics are different from the existing traditional distributed real time systems.

X. Kong et. al. [5, 6] presented a model for virtual infrastructure performance and fault tolerance. But it is not well suited for the fault tolerance of real time cloud applications.

For the non-cloud applications, a baseline model for distributed RTS is, distributed recovery block [7] proposed by

K. H. Kim which is very basic in nature.

Another model is "time stamped fault tolerance of distributed RTS" [8], which is proposed by S. Malik and M. J. Rehman. This model incorporated the concept of time stamping with the outputs.

Adaptive Fault Tolerance in Real Time Cloud Computing AFTRC [9] by Sheheryar Malik and Fabrice Huet., 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.

Our proposed model is based upon adaptive reliability assessment of virtual machines in cloud environment depending on the time taken, the resource availability and the availability of processing capacity techniques. The fault tolerance in this paper is done by calculating the reliability of virtual machines using the available weightage of reliable factors (RF) i.e. memory and cpu availability.

III. Proposed model

A scheme is devised here which is of pro-active fault tolerance for the applications running on cloud infrastructure. The model name is Weighted RF based Fault Tolerance for Cloud Computing. This scheme tolerates the faults on the basis of reliability of each computing node, i.e. virtual machine. A virtual machine is selected for computation on the basis of its reliability. The model is shown in figure 1.

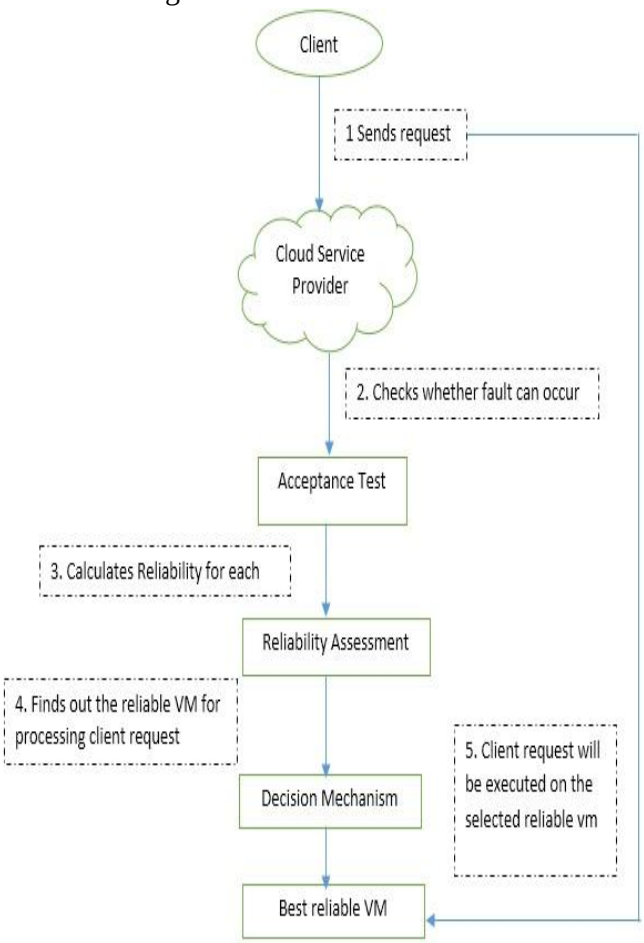

Fig 1. Proposed model

The reliable VM is identified based on the time, previous history, resource availability and available processor capacity. The propose method use two parts. One is the set of virtual machines and other is adjudication node such as the main server. The adjudicator contains the time checker, reliability assessor and decision making algorithms to find the reliable VM. After process, reliable VM is identify to process the client request. A virtual machine is selected for computation on basis of reliability and can be removed, if does not perform well.

In this scheme, we have 'N' virtual machine, which run the 'N' variant algorithms. Algorithm 'X1' runs on 'Virtual machine-1', 'X2' runs on 'Virtual machine-2', up till 'Xm', which runs on 'Virtual machine-m'. Then, we

have TC module which checks the timing of each result. On the basis of the timing, resource and processing power availability, the RA module calculates and reassigns the reliability of each module. Then all the results are forwarded to DM module which selects the output on the basis of best reliability. The output of a node with highest reliability is selected for the further incoming task.

A. Working

As State above this technique has M nodes (virtual machines). Each node is taking input data from the input buffer. This input is concurrently passed to all the virtual machines, which run diverse software. Working of each module is as given below.

Acceptance Test: Check whether the fault can occur or not. Here the fault is a failure of VM or files stored is in corrupted form or not. The acceptance test can respond to both success and failure case of the VM. If the VM is failed, then that VM is not considered. And if, the VM is not failure but the data is corrupted then, that VM is considered for time checker.

Time Checker: Time consuming is calculated in milliseconds. It will monitors the timing of results produced by each module. TC module passes the results to RA (reliability assessor) module. It only passes the correct results of those nodes which produces the result before deadline time. It also informs the RA module to compute the new reliabilities of all the nodes. If the current tasks on the VM are executed within the deadline than the result of TC is further passed to RA module.

Reliability Assessor Module: 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 result within the time, its reliability decreases. The reliability assessment algorithm is more convergent towards failure conditions.

Here, reliability of each virtual machine is calculated on the basis of the weightage of the following reliable factors:

- Memory Availability
- CPU Availability
- Previous History

Memory Availability: Memory is taken as resource. The memory availability for each VM is considered separately. If the tasks running on each VM has been completed within given deadline then, the memory availability for each VM is determined at each cycle. So that, for the upcoming task least utilized VM can be selected to avoid fault.

CPU Availability: Here, the processing capacity of each VM is determined at each cycle. If the tasks running on each VM has been completed within given deadline then, the nodes available processing capacity is determined to calculate the reliability of virtual machines.

Previous History: All the previously calculated reliability of VM are stored in the server database. For calculating the reliability of each VM, an average of the available memory and processing power capacity which was determined during each cycle is taken.

Decision Making Module: Depending on the previous history, weightage of available memory and processing capacity, a virtual machine with the highest reliability value is selected for the as the best VM among all for further processing. Hence, a decision for the best VM is selected on the basis of the correctness of previously executed task and its utilization power.

IV. Experiments and results

We have conducted an experiment using CloudSim 3.0.3 and Netbeans. In this experiment, we configured three virtual machines VM1, VM2, VM3. Each virtual machine runs a diverse algorithm, the algorithm has 5 computing cycles. Then we have an adjudication node, which is sending the input data and receiving the results from the virtual

machines to be processed. The VM1 undergoes acceptance test for checking the availability of the data. The same process is done in VM2 and VM3. After processing this mechanism the result is passed to time checker module. In this module the time consumption for each VM is retrieved and passed these result to the reliability assessor module. The reliably 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 and available processing capacity.

In this experiment, we have an input buffer which provided input to all VM. At VM1, it first undergoes acceptance test, which check where there is any fault in VM or not. After that, at VM1, algorithm 1 starts its first task. Further, the time checker module monitors the time taken by each task for its completion. The process runs for 5 cycles. Same procedure was applied on VM2 and VM3 also.

Time checker module checked the timeliness of each virtual machine result and then passed the result to the reliability assessor module. Here at reliability assessor module reliability is calculated at each cycle depending on the availability of memory i.e. resources and the processing capacity.

Reliability Assessment Algorithm

Begin
Input NumofVMms, NumofTasks

Where,
i = i^{th} cloudlet, j = j^{th} vm,
R_{ij}= reliability of i^{th} cloudlet of j^{th} vm,
R_j = reliability of j^{th} vm

For each task (C_{ij})
{
if (Timer_Status=Success)
{
CP_{ij}= getavailableCPU_Usage ();
MM_{ij}=getavailableMM_Usage ();
r_{ij}= (CP_{ij} + MM_{ij}) / 2;
}
}

For each Vm (V_j)
{
R_j= $\sum_{n=0}^{n=i} rij$ **/ NumofTasks;**
}
End

Decision Making Algorithm

Begin

Input R_j nodeReliability

Best R_j: = find_reliability of node with highest reliability among all VM's

End

V. Evaluation

The Evaluation of three Virtual machine is taken at a time, the memory utilization and the available processing capacity. The time is taken in milliseconds. Here the VM3 has least memory and CPU utilization in this experiment. VM1 is not considered as it has highest CPU usage amongst all. Also, Here VM2 and VM3 has almost same output.

Below figure shows the evaluation report of the VMs in 5 computation cycle based on the memory and processor consumption.

Cycle	VM1			VM2			VM3		
	CPU	Memory	Reliability	CPU	Memory	Reliability	CPU	Memory	Reliability
1	100	26.91	63.455	100	3.63	51.815	100	34.1	67.05
2	79.5	58.99	69.245	100	5.28	52.64	100	62.18	81.09
3	79.23	79.23	79.23	80.14	60.28	70.21	65.25	30.51	47.88
4	50.09	66.73	58.41	70.59	70.59	70.59	59.74	19.48	39.61
5	3.22	3.22	3.22	82.6	88.4	85.5	99.75	99.5	99.625
	Average Reliability		54.71	Average Reliability		66.15	Average Reliability		67.051

Fig 2. Simulation Results

Figure 3 shows the available memory for each VM. Here VM3 contains more memory space, in this situation the VM3 response quickly. So here VM3 is taken as reliable.

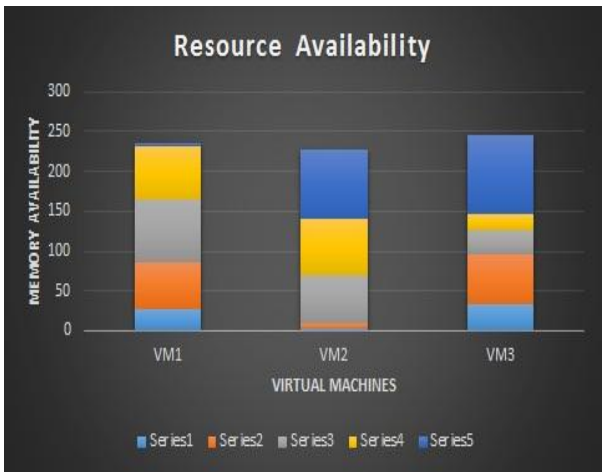

Fig 3. Resource Availability

Figure 4 shows the CPU availability of each VM. Here VM2 has least CPU usage amongst all, i.e. its processor availability is more as compared to other VMs.

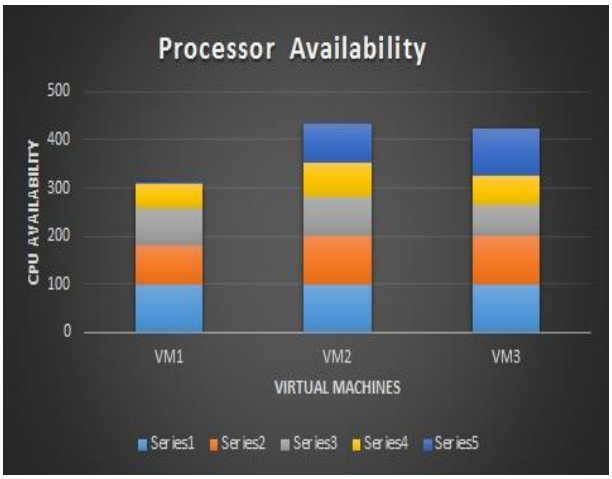

Fig 4. Processor Availability

On the basis of the above results, VM2 and VM3 have almost same reliability, but the result of VM3 will be selected as VM2 has less resource availability as compared to VM3. Also, reliability of VM3 is greater than VM2

VI. Conclusion and future work

Fault Tolerance in Cloud Computing is a very important field of research which needs more efforts in order to come up with a unified plan that would work on all cloud platforms. Use of cloud infrastructure for real time applications increases the chances of errors. As the cloud nodes (virtual machines) are far from the transceiver (job submitting node). Many real time systems are also safety critical systems and which should be reliable. Also, the correctness of result depends not only on the logical result, but also on timeliness and fault tolerance characteristics. So, in order to get correct result, there is an increased need to tolerate fault for such type of systems to be used with cloud infrastructure. Here, with this, in order to tolerate fault, we gave an idea to utilize CPU usage and Memory usage parameters to determine the reliability of VM. The proposed method gives reliability of the Virtual machines, the reliability is taken based on the weightage of available resources of the VM. The time complexity of the process is also minimized because it will not use any backward method. It has a dynamic behaviour of reliability configuration. The reliable VM identification technique used in this process is very efficient to improve the QOS of cloud. In future, our proposed model is aimed at helping the cloud scheduler for the scheduling decisions on the basis of certain network and infrastructure characteristics.

References

[1] Sun Microsystems, Inc. "Introduction to Cloud Computing Architecture" White Paper 1st Edition, June 2009
[2] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in Proceedings of the 1st ACM symposium on Cloud computing, ser. SoCC '10. New York, NY, USA: ACM, 2010, pp. 193–204.
[3] Anju Bala, Inderveer Chana," Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012 ISSN (Online): 1694-0814 www.IJCSI.org
[4] W. T. Tsai, Q. Shao, X. Sun, J. Elston, "Real Time Service- Oriented Cloud Computing", *School of Computing, Informatics and Decision System Engineering Arizona State University USA*, http://www.public.asu.edu/~qshao1/doc/RTSOA.pdf
[5] X. Kong, J. Huang, C. Lin, P. D. Ungsunan, "Performance, Fault-tolerance and Scalability Analysis of Virtual Infrastructure Management System", 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications, Chengdu, China, August 9-12, 2009
[6] X. Kong, J. Huang, C. Lin, "Comprehensive Analysis of Performance, Fault-tolerance and Scalability in Grid Resource Management System", *2009 Eighth International Conference on Grid and Cooperative Computing*, Lanzhou, China, August 27-29, 2009
[7] K. H. Kim, "Structuring DRB Computing Stations in Highly Decentralized Systems,: *Proceedings International Symposium on Autonomous Decentralized Systems*, Kawasaki, 1993, pp. 305-314
[8] S. Malik, M. J. Rehman, "Time Stamped Fault Tolerance in Distributed Real Time Systems"; *IEEE International Multitopic Conference, Karachi, Pakistan*, 2005
[9] Sheheryar Malik and Fabrice Huet,"Adaptive Fault Tolerance in Real Time Cloud Computing" - IEEE World Congress on Services 2011

[10] 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)
[11] 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.
[12] ZibinZheng, Tom Chao Zhou, Michel R. Lyu, and Irwin king "FT-Cloud: A Component Ranking Framework for Fault-Tolerant Cloud Applications "2010 IEEE 21st International Symposium on Software Reliability Engineering.

