Volume 5, Issue 6, June 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Enhanced Adaptive Fault Tolerance Model for increasing Reliability in Cloud Computing

Pawandeep Kaur **Manpreet Kaur**

Abstract— Today's world is an era of cloud computing. Cloud computing aim to power the data centers as a network of virtual services like hardware , database, user-interface etc so that user can access information and applications via internet from anywhere in the world on demand at competitive costs depending on users quality of service requirements. Real time system is a system in which computation guarantees strict time constraints. In a real time cloud applications computations is done at remote cloud computing nodes which may sometimes result in chances of error due to a loose control over computing nodes. This system should be highly reliable so there is an increased demand of fault tolerance to achieve reliability for real time computing on cloud infrastructure. In this paper, a fault tolerance model for real time computing is proposed. The proposed model, is a model named Enhanced Adaptive Fault Tolerance model for increasing reliability in Cloud Computing. The system tolerates the faults and makes the decision on the basis of reliability of the processing nodes that is virtual machines. If a correct result within the time limit will be produced by a virtual machine increases. And if it fails to produce correct result within time limit then its reliability decreases. This model have RAM (Resource Allocation Module) and User Controlled Exception Handling which make it more elastic to the environment. The proposed technique is based on the execution of task if the task is exceeding the limit then it send the task to exception handling section, but it also checks that if the task is about to finish and required the time that is sustainable then it provide the elasticity to that task. The virtual machine instances can be of same type or of different types. The main essence of the proposed technique is to provide elasticity to the task and then admin will decide whether to resubmit the task of same virtual machine or to reject that task. The resubmitted task will be computed again and assign new reliability value

Keywords— cloud computing, reliability , fault tolerance

I. INTRODUCTION

Cloud computing is the access to services of a computers via the Internet or a local area network. Cloud user request this access from a set of web services that manage a pool of computing resources (i.e., machines, network, storage, operating systems, application programs) and user's request granted this pool of resources until he or she releases them. It is called "cloud computing" because the user cannot actually see the physical location and organization of the equipment hosting the resources they are ultimately allowed to use. Virtualization technique is common in cloud computing, that is., many virtual machines even with different operating systems may be running in a single physical machine. The NIST cloud computing definition **"**Cloud computing is a model for enabling convenient and on demand network access to a shared pool of computing resources (e.g., servers, storage, networks, applications, and services) that can be rapidly released with minimal management effort or service provider interaction.". [9].

Web-based email services like Gmail , yahoo mail and Hotmail deliver a cloud computing service: users can access their email "in the cloud" from any computer anywhere in the world with an Internet connection. The emails are hosted on Google's and Microsoft's servers, rather than being stored locally on the client computer. Many other examples of cloud computing are like social applications (e.g., Facebook, Twitter), VoIP (e.g., Skype, Google Voice), media services (e.g., Picassa, YouTube), content distribution (e.g., BitTorrent).

Fault Tolerance in Cloud Computing is the ability of a system to respond gracefully to an unexpected hardware or software failure. Fault tolerance in cloud computing platforms and applications is a crucial issue. Fault tolerance in cloud computing is one that can continue to correctly perform its task in the presence of hardware failures or software or to operate satisfactorily in the presence of faults. The main benefits of implementing fault tolerance in cloud computing include failure recovery, lower cost, improved performance of a system by making it highly reliable. The backbone of software reliability is avoiding the faults. Since cloud architecture is very complex and built on data centers comprising thousands of interconnected servers with capability of hosting a large number of applications and distributed globally so fault avoidance techniques or fault removal techniques such as testing to detect and remove fault, therefore, won't be so easy enough in the case of cloud computing. In this context to achieve reliability to a greater extent, the system must be fault tolerant. Fault tolerance is required to make system highly reliable and to increase efficiency of a system.

II. RELATED WORK

Fault tolerance methods come into play the moment a fault enters the system boundaries. So fault tolerance Techniques are used to predict these failures and take an appropriate action before failures actually occur.

Malik and Huet [1] proposed a model Adaptive Fault Tolerance in Real –time Cloud computing based upon adaptive reliability assessment of VM in cloud environment.. Real time systems are safety critical so a need of reliable and high fault tolerance model of real time computing is discussed. The proposed technique is based on the execution of design diverse variants on multiple virtual machines, and assigning reliability to the results produced by variants. Jahwar and Piuri [3] also discusses need of fault tolerance for the better execution of cloud computing. They proposes an innovative, system-level, modular perspective on creating and managing fault tolerance in Clouds, by means of a dedicated service layer.

Ganawi and Hellerstel [2] proposes a new type of cloud service Failure as a Service (FaaS), which allows cloud services to routinely perform large-scale failure drills in real deployments. FaaS enables cloud services to routinely exercise large-scale failures online, which will strengthen individual, organizational, and cultural ability to anticipate, mitigate, respond to, and recover from failures. Bala and Chana [4] states that in order to minimize failure impact on the system and application execution, failures should be anticipated and proactively handled. .The fault tolerance techniques covering its research challenges, tools used for implementing fault tolerance techniques in cloud computing. Autonomic fault tolerance is implemented dealing with various software faults for server applications in a cloud virtualized environment. When one of the servers goes down unexpectedly, connection will automatically be redirected to the other server. Data replication technique is implemented on virtual machine environment.

Very few has introduced the fault tolerance on the basis of reliability of node and reliability assessment of node. This paper aims to provide a better understanding of fault tolerance techniques by providing a model used for fault tolerance in cloud environments . So in this proposal we have added RAM (Resource Allocation Module) and User Controlled Exception Handling to the model to make it more elastic to the environment. And we also extended time checker with an elasticity constraint which will provide time checker more effective result.

III. PROPOSED MODEL (EAFTRC)

A fault tolerance scheme running on cloud infrastructure called **E**nhanced **A**daptive **F**ault **T**olerance model for increasing **R**eliability in **C**loud **C**omputing. This model is an enhancement in a model name Adaptive fault tolerance in real time cloud computing (AFTRC) [1]. 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 major problem in Adaptive fault tolerance model is that it is completely relying on the time checker but in order to create proper fault tolerance model we have to consider other parameters for toleration faults in cloud. So in this proposal we have added RAM (Resource Allocation Module) and User Controlled Exception Handling to the model to make it more elastic to the environment. And we also extended time checker with a elasticity constraint which will provide time checker more effective result.

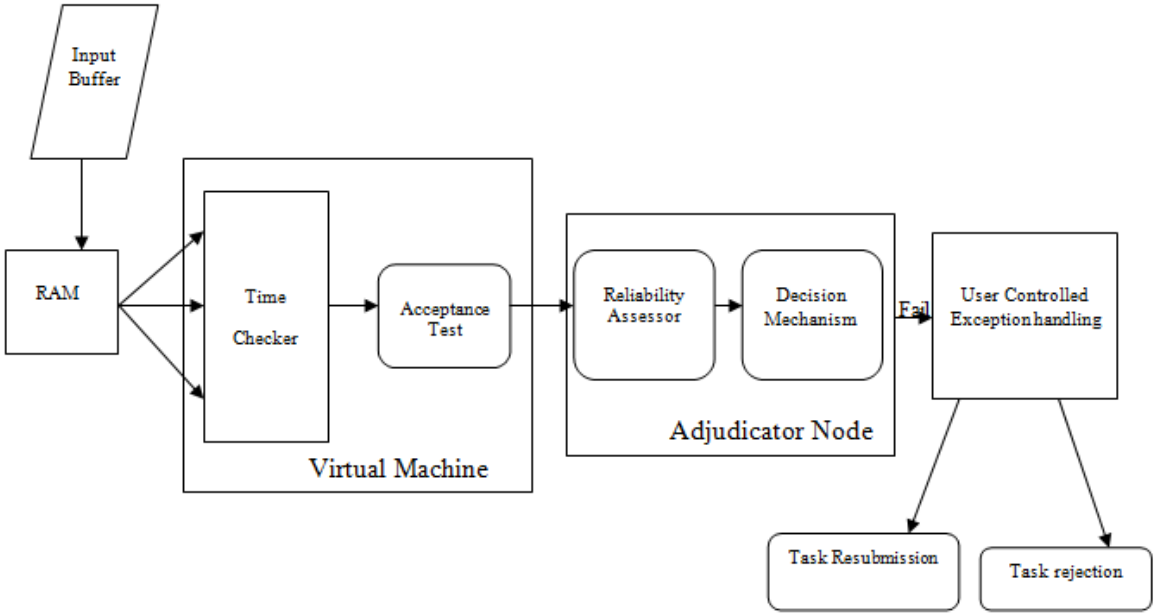

Figure 1: Enhanced Adaptive fault Tolerance model for increasing reliability in cloud computing

In enhanced adaptive fault tolerance model for increasing reliability, we have two main types of node, one is virtual machines, running on cloud infrastructure and the other is the adjudicator node. Virtual machine contains time checker which monitor the execution of the task and an acceptance test for its logical validity. Second type of node that is adjudicator node contains Reliability Assessor and Decision Mechanism. Adjudicator node can be run on cloud infrastructure and on user infrastructure which depends on type of real time applications , It can be placed near to the sensors or submission node.

A. WORKING OF A MODEL

In this scheme, we have 'N' number of virtual machine, which run 'N' different tasks depend upon requirement of resources taking input data from an input buffer.

Input Buffer :- Input Buffer has a responsibility to give input data (required resources) to each and every virtual machine. The input is concurrently passed to all the virtual machines. Each virtual machine takes the input from input buffer, executes it and produces result. Different tasks are run on each module to avoid the chances of faults by coincident.

RAM(Resource allocation module) :-RAM Module takes input from input buffer that is tasks(required resources). According to resource requirement RAM suggest VM(Virtual Machine) to the task which is finally selected by Admin of the cloud.

Time Checker :- Time Checker module is a time checking unit which monitor the execution of task that is result produced by each task . it monitor the execution of task if the task is exceeding the limit (produces the result after deadline time) then it send the task to exception handling section , but it also checks that if the task is about to finish and required the time that is sustainable then it provides the elasticity to that task . TC module passes the result to Acceptance test module .

Acceptance test :- Acceptance test module is provided to each virtual machine . It performs the acceptance test for each and every task output and also verifies the correctness of the result produced by a task on the same virtual machine. If the result is correct then it passes the result is correct then it passes the result to Reliability Assessor for checking its reliability otherwise if result produced is incorrect then it does not pass the result to RA module , it sends a Validity exception signal to TC.

Reliability Assessor :- The Reliability Assessor module is main core module of this model and responsibility of this module is to calculate the percentage of reliability for each virtual machine . The proposed system tolerates the faults and makes the decision on the basis of reliability of the virtual machine. The reliability of a virtual machine depends upon the three factors which are as following :- cloud failure , host failure and incorrect value assigned to the virtual machine at the time of creation . If a cloud failure occurs then there will be a 33% decrease in the reliability . If due to some reason host fails then there will be a fall of another 33% in reliability. If the value assigned to the variable of a virtual machine is incorrect at the time of creation then the reliability of that virtual machine will be zero , so reliability of each virtual machine will be calculated by the reliability assessor and then according to this value of reliability calculated decision mechanism module will perform its function.

Decision mechanism(DM) :- Decision mechanism module selects the output from the reliability assessor (RA) module. According the reliability calculated by the reliability assessor decision mechanism will take its decision. The best virtual machine will be decided on the basis of reliability of virtual machine. Higher the reliability better will be the virtual machine . if the reliability of two virtual machine will be same then decision will be taken on the basis of value entered or input given to the virtual machine . Virtual machine whose input was given earlier than the other one that machine will be considered as best virtual machine otherwise virtual machines having highest value of reliability will be decided as best virtual machine. After performing its responsibility DM module passes the output to the User Controlled Exception Handling section.

User Controlled Exception Handling section :- In this User Controlled Exception Handling section , time checker module sends the task by monitoring the execution of task that is if the task is exceeding the limit then it send the task to exception handling section, but it also checks that if the task is about to finish and required the time that is sustainable then it provide the elasticity to that task. Elasticity is provided to the task in this section and admin decide whether to resubmit the task of same virtual machine or to reject that task. If the task was resubmitted then elasticity will be provided to that task and again that task with same virtual machine will follow each step and calculate new reliability value. Decision mechanism will consider that virtual machine as best VM. If the task was rejected then reliability of that virtual machine whose task was rejected was decreased by 33% and another task were executed and best virtual machine will be decided accordingly.

B. FAULT TOLERANCE MECHANISM

We apply reliability assessment algorithm for each virtual machine one by one . Initially declare three variables i , c, m of type input, cloudlet, virtual machine. Initially the value of i is zero then iterate over the cloudlet list, for the no of elements present in the list with i as a counter.. VmId is a Id of a virtual machine in which tasks are running. Comparing sta property of c with a constant 4, and if it is greater than constant than, fetch VmId, and use it as index to fetch the element from Vmlist (virtual machine list) present at that index and assign it to m . The reliability of a virtual machine will be compared to check failed cloud if any possible and then increment fail cloud property of virtual machine variable m by one. rf is a reliability factor which increases the reliability of a virtual machine. Reliability increases by a factor of 33.3. check if no of processors is less than datacenter and check mips is less than equal to 500 (initially assigned value) then fetch host list and set reliability is equal to 33.3. If failed host is equal to zero then reliability will be incremented by value 33.3 and similarly if failed cloud is equal to zero then reliability will again incremented by a factor 33.3. Else reliability is equal to zero when if condition is not satisfied.

Reliability Assessment Algorithm

Begin

Input *i , Cloudlet c ,Virtual machine m*


```
        for(i=0;i<cloudletList.size();i++)
              { c=cloudletList.get(i)
      if(c.sta > 4)
    { m=vmlist.get(c.getVmId());
   m.failCloud=m.failCloud+1; }    }
   Vm v;
   for(i=0;i<vmlist.size();i++)
   { v=vmlist.get(i);
   if(v.persons < datacenter0.getHostList().size()*2 && v.mipscheck<=500)
   {   System.out.println("mips="+v.mipscheck);
   v.rf=1;
   v.relaiblity=33.3;
   if(datacenter0.failedhosts==0)
   { v.rf=v.rf+1;
         v.relaiblity=v.relaiblity+33.3;   }
              if(v.failCloud==0)
   { v.rf=v.rf+1
    v.relaiblity=v.relaiblity+33.3 ;}}
         else
         {  v.rf=0;
         v.relaiblity=0;}
```

Decision Mechanism Algorithm:

In a Decision Mechanism Algorithm defined a variable best of type Vm initialize it with the first element of the vmlist (virtual machine list). We are iterating over the vmlist which is a list containing objects (tasks) of virtual machine Vm for the no of elements present in the list with I as a counter . Defining the variable v of type Vm and initialize it by fetching the i^{th} element from the vmlist. Compare reliability factor rf that is if rf value of v is greater than that of best , then set v as best. Virtual machine with highest reliability will be considered as a best virtual machine by a decision mechanism module.

```
Begin
Declare virtual machine v
Vm best=vmlist.get(0)
    for(i=0;i<vmlist.size();i++)
         v=vmlist.get(i);
      if(v.rf>best.rf)
        best=v;
End
```

C. DIFFERENT SITUATIONS

Due to the complexity of software and different parameters there could be different situation leading to success and failure of results.

Situation 1: Complete Failure Free:-All the task on each virtual machine produce the correct result . Time Checker passes the result to Acceptance Test as results are produced before time overrun. Reliability Assessor computes the new reliability to each virtual machine and Decision Mechanism takes an output from Reliability Assessor and considers the best virtual machine. It is a complete failure free situation as there is no elasticity provided and exception does not occur in any virtual machine.

Situation 2: Partial Failure:- cloudlet or task failure , In this case RAM assigns virtual machine to the task and then Time Checker finds output within the time limit and fails to satisfy the time limit then reject condition occurs for that task and that task will be failed. Next task from cloudlet list will be executed and pass through all modules and find correct result

Situation 3: Complete Failure:- Incorrect Values assigned , In this case, the incorrect values assigned to the variables of a virtual machine so Time Checker fails to find output within the time limit. Acceptance Test does not receive a correct result to verify result, then Reliability Assessor computes zero reliability value for that virtual machine whose variables had assigned incorrect values. Virtual Machine with zero reliability will be considered worse and another will be considered best virtual machine by the Decision Mechanism. RAM passes the task to another virtual machine.

Experiments:- We have created two virtual machines and then run our tasks on them. There are no of tasks running on each virtual machine and each of these tasks are performed in one computing cycle. Each virtual machine runs a different task. RAM (Resource Allocation Module) receives the data from input buffer. Time Checker monitors the execution of a task and Acceptance test verifies the result of each task in a virtual machine. Then we have an adjudication node, which is sending the input data and receiving the results from the virtual machine to be processed. Adjudication node contains Reliability Assessor and Decision Mechanism. Our experiment has the following Table I shows us the description of virtual machine for proposed model.

TABLE I : VIRTUAL MACHINE DESCRIPTIONS		
	Virtual machine-1	**Virtual machine-2**
VM id	0	1
Mips	500	500
Memory(RAM)	2048(MB)	2048(MB)
Bw	1000	1000
No of Processors	1	2

Vm represents a VM: it runs inside a Host, sharing host List with other VMs and then It processes Cloudlets. This processing happens according to the policy, defined by the Cloudlet Scheduler. Each VM has a owner, which can submit cloudlets to the VM to be executed .The second VM-2 (Virtual Machine) will have twice the priority of VM-1 and so will receive twice CPU time.

Adjudication Node:-

There is an adjudication node , which contains the following modules:

- Reliability Assessor
- Decision Mechanism

In this experiment, we have an input buffer which provides input to RAM. RAM suggests VM to the task . At virtual machine -1 the task was passed to TC module for the verification of timeliness and also to check elasticity provision. The result of TC module is passed to Acceptance Test, which verifies the correctness of its result. The process runs for each cycle.

At virtual machine -2 , same procedure was applied to produce result for the task in each computing cycle. TC passes the result upon verification of timeliness and elasticity provision to acceptance test for the verification of result. The same process runs for each cycle.

RAM module assigns VM (virtual machine) to every task and then each task was processed through TC for the verification of timeliness and to check whether elasticity is needed or not. TC passes the result to Acceptance Test for the verification of result .then the result of the task was passed to the RA module for assigning and updating the reliability of each virtual machine. RA module passed the result to the decision mechanism which selected the result with best reliability. DM determines the best virtual machine among all. User Controlled Exception Handling section, time checker module sends the task by monitoring the execution of task that is if the task is exceeding the limit then it send the task to exception handling section, but it also checks that if the task is about to finish and required the time that is sustainable then it provide the elasticity to that task. Elasticity is provided to the task if needed in this section and admin decide whether to resubmit the task of same virtual machine or to reject that task.

In Table II , In Cycle 1 input values are entered in User interface for cloudlet's length , cloudlet's file size and cloudlets output size. Reliability has been calculated by reliability assessor with status success and elasticity was not needed for both virtual machine-1 as well as virtual machine-2 .In this model elasticity was provided from 100 to 105 ms so in cycles 2 to 4 elasticity was provided in case of virtual machine-1 and not in case of virtual machine -2. In case of cycle 5 the values of cloudlet's exceed the limit and in this case time checker now passes the result to user controlled exception handling section and admin will decide whether to resubmit this task again or to reject the task. In case of cycle 5 where the same task was resubmitted by the admin, there new reliability value will be calculated by the reliability assessor with status created shown. When the task was rejected by the admin then its reliability will decrease by factor 33.3 and the best virtual machine in this case considered being a virtual machine 2. In cycle 5 and 6 there is no need of elasticity in case of virtual machine 2 but elasticity will be provided in case of cycle 7.

TABLE II : EXPERIMENT RESULTS FOR RELIABILITY ASSESSMENT													
Cycle	**Input**				**Virtual Machine-1**				**Virtual Machine-2**				
	Cloudlet's Length	**Cloudlet's File Size**	**Cloudlet's Output Size**	**User Choice**	**Status**	**Elasticity**	**Reliability**	**Time**	**User Choice**	**Status**	**Elasticity**	**Reliability**	**Time**	**DM (best Vm)**
1	10000	10000	10000	------	Success	No	99.89 %	20	------	Success	No	99.89%	40	Vm1
2	25600	25600	25600	------	Success	Yes	99.89 %	102.4	------	Success	No	99.89%	51.2	Vm1
3	26220	26220	26220	------	Success	Yes	99.89 %	104.88	------	Success	No	99.89%	52.44	Vm1

4	26225	26225	26225	------	Success	Yes	99.89%	104.9	------	Success	No	99.89%	52.45	Vm1
5	26230	26230	26230	resubmit	created	No	99.89%	-----	------	Success	No	99.89%	52.46	Vm1
6	26230	26230	26230	reject	canceled	No	66.6%	-----	------	Success	No	99.89%	52.46	Vm2
7	52450	52450	52450	reject	canceled	No	66.6%	-----	------	Success	yes	99.89%	104.9	Vm1
8	52500	52500	52500	resubmit	created	No	99.89%	-----	resubmit	created	No	99.89%	-----	Vm1
9	52500	52500	52500	reject	canceled	No	66.6%	-----	reject	canceled	No	66.6%	-----	Vm1

IV. CONCLUSIONS AND FUTURE WORK

We presented an approach for fault tolerance on the applications in virtual machine instances. The proposed approach takes an advantage of providing elasticity to a task running in a virtual machine .The system checks that if the task is about to finish and required the time that is sustainable then it provide the elasticity to that task. This scheme is highly fault tolerant. In our experiment, we have used two virtual machines in a cloudsim. In this model reliability assessment of all the virtual machines is done and the best VM is selected by the Decision mechanism. It is done at every task before and after execution.The future work focuses on finding more parameters and adding to this scheme so that our system should be more faults tolerant.

REFERENCES

[1] Sheheryar Malik, Fabrice Huet "Adaptive Fault Tolerance in Real Time Cloud Computing", Research Team OASIS INRIA - Sophia Antipolis 06902 Sophia Antipolis, France, IEEE World Congress on Services,2011

[2] *Haryadi S. Gunawi, Joseph M. Hellerste l"*Failure as a Service (FaaS): A Cloud Service for Large-Scale, Online Failure Drills" Electrical Engineering and Computer Sciences University of California at Berkeley, Technical Report No. UCB/EECS-2011-87, July 28, 2011

[3] Ravi Jhawar, Vincenzo Piuri," Fault Tolerance Management in Cloud Computing: A System-Level Perspective" *Graduate Student Member, IEEE,2012*

[4] Anju Bala, Inderveer Chana, "Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing", Science and Engineering Department, Thapar University Patiala-147004, Punjab, India, IJCSI International J Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012

[5] R.Senthil Kumar, K.K.Kanaga Mathan Mohan, "E ffective Failure prediction for Meeting Soft Deadlines Using Resubmission Impact in cloud", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 5, May 2013

[6] Bingpeng Zhu1, Gang Wang, "Proactive Drive Failure Prediction for Large Scale Storage Systems", Nankai-Baidu Joint Lab, College of Information Technical Science, Nankai University, Tianjin, China,IEEE,2013

[7] Matei Zaharia, Mosharaf Chowdhury," Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing" University of California, Berkeley

[8] Lee Badger, Robert Bohn, NIST US Government Cloud Computing Technology Roadmap, Release 1.0 (Draft)" Cloud Computing Program Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899 November 2011

