ISSN: 0976-3104
ISSUE: COMPUTER SCIENCE

ARTICLE

ALGORITHMS TO ENHANCE THE RELIABILITY OF VIRTUAL NODES USING ADAPTIVE FAULT TOLERANCE TECHNIQUES

Mridula Dhingra*, Neha Gupta

Faculty of Computer Applications, Manav Rachna International Institute of Research and Studies, Faridabad, INDIA

ABSTRACT

Cloud computing has various benefits in terms of cost saving, flexibility, efficiency etc. but it is not much efficient to perform real time computing in cloud infrastructure. The real time system acquires benefits of scalable virtualized environment and intensive cloud computing capabilities for executing real time tasks. Most of the real time applications undergo its processing on remote cloud computing nodes and chances of error occurrence is also high due to undecided latency. And it is essential that the safety critical applications should have higher reliability. To determine whether the fault tolerance mechanism is having the higher reliability and availability or not, an efficient adaptive fault tolerance mechanism has been proposed through reliability assessment architecture to enhance the reliability of the system. Also fine grained check point algorithm is used to minimize the latency.

KEY WORDS
Cloud Computing, Virtualization, Fault Tolerance, Reliability Assessment, Fine grained checkpointing.

Received: 3 Aug 2019
Accepted: 15 Oct 2019
Published: 20 Oct 2019

***Corresponding Author**
Email:
mridula.fca@mriu.edu.in
Tel.: +91-9871462056

INTRODUCTION

Now a days Cloud technologies have become a vital development in the era of information technology. Like every technology, cloud computing undergoes from some severe issues like fault tolerance of real time systems [1]. Virtualization and Internet-based Cloud computing leads to different kinds of failures to arise and hence, leads to the requirement for reliability and availability has turn out to be an essential issue. To ensure reliability and availability of cloud technologies, a scheme for fault tolerance need to be developed and applied. Majority of the early schemes for fault tolerance were paying attention on only one way to tolerate faults. Now cloud computing has changed the trend that how computing will perform, But the users are still adamant to work for real time applications. But many researchers are now working on real time applications. Real time systems are scrutinized on the basis of timeliness and fault tolerance factors[2]. Timeliness is the capability of a real time system to complete the proposed assignment within a specific time and fault tolerance is the ability of the system to work elegantly in the occurrence of fault. Now a days as the requirement of cloud computing is growing day by day, real time systems can be carried out on cloud environment. In most of the real time cloud applications, execution is done on remote cloud computing nodes. So the probability of errors become high because of undecided latency of computing node but on the contrary most of real time systems are safe and extremely reliable and to conquer the reliability of real time systems in cloud computing, there is huge demand of fault tolerance amongst users. [3] Cloud infrastructure has established many issues related to real time computing. In cloud computing, the latency of virtual nodes is not known. Even if one resolves the latency, but it amends after some period of time. In this research paper, this constraint has been overcome by proposing a framework based on adaptive reliability assessment and fault tolerance of VMs in real time applications.

To attain reliability the cloud computing, vendors implements a variety of fault tolerance methods to run the system in case of faulty conditions [2].

Need of Fault Tolerance Methodology

- Fault tolerance provides the method or module to do the task effortlessly even though in the incidence of malfunction.
- Fault tolerance assures quality of service for users by definite recital.
- User can rearrange and append software instructions in accordance with their needs.
- Fault tolerance is a technique or concept to implement a system to perform well in unexpected situations, e.g. when a part of cloud is not working up to the level of satisfaction the whole server of the cloud will not collapse.

RELATED WORK

Limam and Belalem (2014) [4] proposed an adaptive checkpoint method in which the methodology to eliminate preventable checkpoints and adding of additional checkpoints was suggested. This methodology works on the current status of various cloud components. This method either enhances or reduces the intervals between checkpoints.

Cao et al (2015) [5] have presented a scheme to support energy jobs and jobs with priorities. This method is also based on checkpointing.

Bobroff et al (2007) [6] have discussed the methods to forecast the workload of VMs and based on this workload classification of VMs is done. Authors have proposed a VM management algorithm which reduces the total power consumption on run time without compromising the SLA violation rates.

Verrna et al (2008) [7] have proposed a model based on VNs. Their methodology works on the assumption that no multiple copies of VM exist. The proposed method is based on a power-aware VM placement controller to control the heterogeneous servers and VMs.

Proposed work

In [Fig. 1], the proposed reliability assessment architecture is discussed. The proposed architecture will consist of N virtual nodes. Each node will take the input data from the input buffer and then the input will be given to the each virtual machine. Every time the input will be taken by the node, it will perform the operation and provide the result. The proposed reliability framework will primarily divided into three modules, these are:

- Acceptance Computation
- Reliability Assessment
- Fine-grained checkpointing
- Decision Mechanism
- Replication

Decision Mechanism and Replication will be discussed in next upcoming research papers.

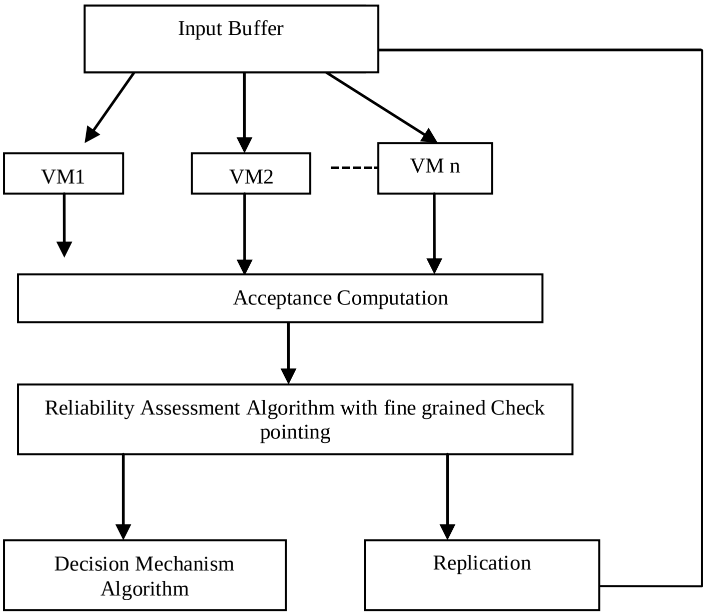

Input Buffer
VM1
VM2
VM n
Acceptance Computation
Reliability Assessment Algorithm with fine grained Check pointing
Decision Mechanism Algorithm
Replication

Fig. 1: Reliability assessment architecture.

Acceptance computation (AC): The proposed scheme will utilize n Virtual Machines wherein each machine acquires its input from input buffer. All virtual machines concurrently acquires input from it. Every single machine takes the input, executes the application and produces result. Then the results are directed towards the Acceptance Computation module. If the acquired results are correct, then its computational time is compared with threshold time. If the computational time exceeds threshold time, the node is considered as a failure node and its reliability is not tested and backward recovery will be performed through fine-grained checkpointing, otherwise the results will passed to reliability assessment module to measure the reliability of virtual machines.

Reliability assessment (RA): The reliability is a continuous measure and changes its value after every computation cycle. By default, initial value of reliability of each virtual node is 100 %. In this module reliability assessment algorithm (proposed) will be used to measure the reliability of virtual machines. However, if the reliability value of virtual machine will be less than the minimum reliability then the backward recovery will be performed by the system through fine- grained check pointing.

The proposed algorithm reliability assessment algorithm will depict the minimum and maximum reliability of nodes by identifying and replacing the failure nodes. The node status is checked for reliability analysis to fix the specific nodes for execution by eliminating the failure nodes. Step 3 shows node status, if the node status is passed it proceeds the process of IR= IR+(IR * RF).

```
Reliability Assessment Algorithm

Input :Initially reliability ← IR=1, n :=1  from configuration RF,
Step1: MR ←maxReliability
Step2: Mir←minReliability
Step3: Ns ← node status
Step4: If Ns = pass then
Step5: IR= IR+(IR * RF)
Step6: If n > 1 then
Step7: n=n-1
Step8: End if
Step9: Else if Ns = Fail then
Step10: IR =IR –(IR*RF*n)
Step11: n=n+1
Step12: End else if
Step13: If IR >= MR then
Step14: IR = MR
Step15: End if
Step16: C ← count=1;
Step17: While c<=4
Step18: C+1
Step19: If IR < Mir then
Step20: Ns = dead
Step 21: call_proc: remove_this_node
Step 22: call_proc: add_new_node
End if
End while
```

Virtual machine 0 to 9 is given in [Table 1]. The total time and reliability time is checked for each virtual machine to identify whether the machine performing its task with expected reliability time. In the table it is clearly shown that the 7, 8 and 9 failed to perform the task when its reliability time increases.

Table 1: Reliability time assessment

VMID	Total time	Reliability Time	Result
0	852	0.99734	Pass
1	114	1.994	Pass
2	366	2.992	Pass
3	954	3.9895	Pass
4	776	4.9869	Pass
5	407	5.984	Pass
6	747	6.981	Pass
7	562	7.979	Fail
8	134	8.976	Fail
9	137	9.9739	Fail

Fine-grained check pointing

The proposed fine grained Check pointing algorithm will work on the assumption that the length of the check pointing interval is not fixed and is calculated at the time of present checkpoint. This calculation is based on the failure history of VMs. If the failure rate is less, then the interval between checkpoints is reduced and if the node is having poor failure history, the interval between checkpoints is increases in the opposite scenario.

```
Fine –grained check point algorithm
Input: R←Reliability time
VM ←Virtual machine
T ←thresholds time
NC ← number of count=4
N ← Number count =1
$B_{TF}$←False
$B_{Ts}$←True
Step 1: While (!Task execution completed) do
Step 2: If (R>=T)then
Step 3: Estimation();
Step 4: END IF
Step 5: if(R <= T)
```

IIOAB JOURNAL ISSN: 0976-3104

```
Step 6: R++;
Step 6: N++;
Step 7: If(N<=NC)
Step 8:then    = true
Step 9: update the VM failure information.
Step 10: Create New VM
Step 10: end if
Step 11: end while
```

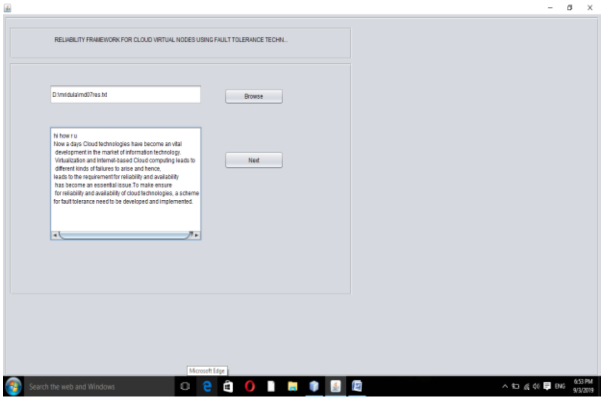
The proposed algorithm is implemented for checking and enhancing the reliability time of VM for completing the particular task efficiently. If the execution of the task halts suddenly, the reliability is checked by relating it with threshold values. If the reliability is greater than or equal to threshold time then the task will be finished within the schedule time without any interruption. If R<= T, the reliability and node count 4 is increased. If the N<=NC thus far, then BTF =true.

Then the VM failure information is updated.

Implementation

In the proposed framework Cloudsim is used to generate and evaluate the results [10]. The classes in the cloudsim package permit the development of algorithms based on fault tolerance which in turn can supervise virtual nodes towards the identification of failures and then later resolves them. This provision can deploy both fine grained check pointing and replication mechanisms. This simulator offers the capability to calculate availability, throughput, time overhead and monetary cost overhead. The timing of result produced by each virtual machine will be monitored with Watch Dog Timer Software. The performance measure was evaluated to prove the efficiency of proposed adaptive fault tolerance mechanism. The performance of the framework was evaluated in terms of execution time. The reliability assessment, decision making and the Fine grained check point algorithm proved their efficiency on the basis of outcome received.

Implementation of the proposed algorithms are shown as follows:



RELIABILITY FRAMEWORK FOR CLOUD VIRTUAL NODES USING FAULT TOLERANCE TECHN...

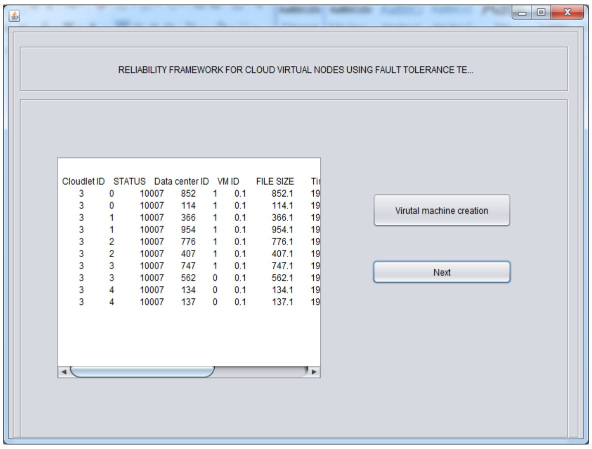
D:\mridula\md07res.txt

```
hi how r u
Now a days Cloud technologies have become an vital
 development in the market of information technology.
 Virtualization and Internet-based Cloud computing leads to
 different kinds of failures to arise and hence,
leads to the requirement for reliability and availability
 has become an essential issue.To make ensure
 for reliability and availability of cloud technologies, a scheme
for fault tolerance need to be developed and implemented.
```

Fig. 2: input buffer

………

In [Fig. 2] reliability framework input has been taken from input buffer, after that virtual machines will be created and input will be given to the virtual machines.



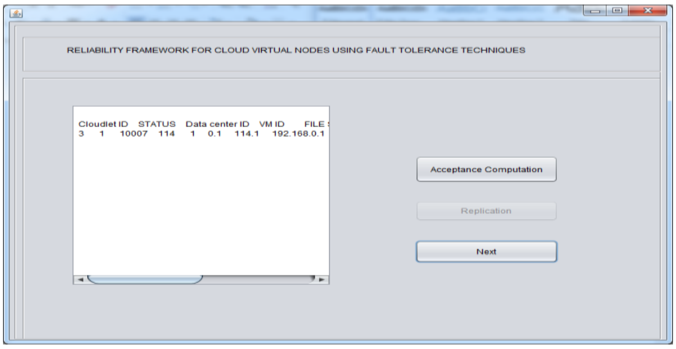
RELIABILITY FRAMEWORK FOR CLOUD VIRTUAL NODES USING FAULT TOLERANCE TE...

```
Cloudlet ID   STATUS   Data center ID   VM ID      FILE SIZE    Ti
    3          0         10007     852     1     0.1     852.1   19
    3          0         10007     114     1     0.1     114.1   19
    3          1         10007     366     1     0.1     366.1   19
    3          1         10007     954     1     0.1     954.1   19
    3          2         10007     776     1     0.1     776.1   19
    3          2         10007     407     1     0.1     407.1   19
    3          3         10007     747     1     0.1     747.1   19
    3          3         10007     562     0     0.1     562.1   19
    3          4         10007     134     0     0.1     134.1   19
    3          4         10007     137     0     0.1     137.1   19
```

Fig. 3: VM creation

………

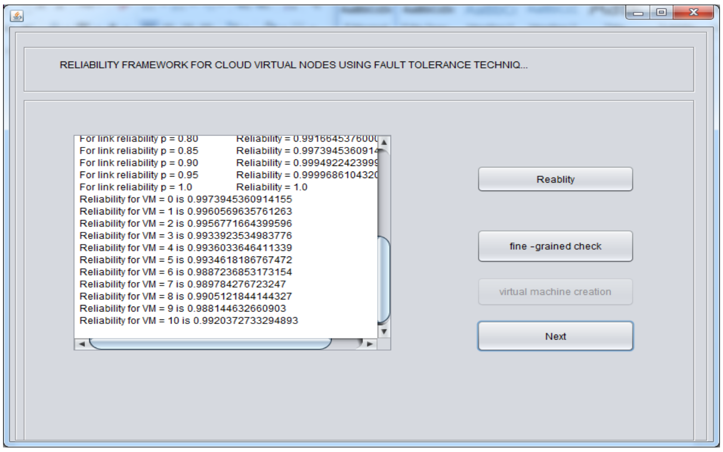
In [Fig. 3] ten virtual machines are created, after creation of virtual machines results will be checked whether it is correct or not.


RELIABILITY FRAMEWORK FOR CLOUD VIRTUAL NODES USING FAULT TOLERANCE TECHNIQUES
Acceptance Computation
Replication
Next

```
Cloudlet ID    STATUS    Data center ID    VM ID       FILE
3     1     10007    114     1    0.1     114.1     192.168.0.1
```

Fig. 4: Acceptance computation

In [Fig. 4], the total time and reliability time is checked for each virtual machine to identify whether the machine performing its task with expected reliability time. During the acceptance computation it is mentioned that the 7, 8 and 9 failed to perform the task when its reliability time increases. To increase their reliability fine-grained check pointing is used.


RELIABILITY FRAMEWORK FOR CLOUD VIRTUAL NODES USING FAULT TOLERANCE TECHNIQ...
Reablity
fine -grained check
virtual machine creation
Next

```
For link reliability p = 0.80          Reliability = 0.9916645376000
For link reliability p = 0.85          Reliability = 0.997394536091
For link reliability p = 0.90          Reliability = 0.9994922423999
For link reliability p = 0.95          Reliability = 0.9999686104320
For link reliability p = 1.0           Reliability = 1.0
Reliability for VM = 0 is 0.9973945360914155
Reliability for VM = 1 is 0.9960569635761263
Reliability for VM = 2 is 0.9956771664399596
Reliability for VM = 3 is 0.9933923534983776
Reliability for VM = 4 is 0.9936033646411339
Reliability for VM = 5 is 0.9934618186767472
Reliability for VM = 6 is 0.9887236853173154
Reliability for VM = 7 is 0.989784276723247
Reliability for VM = 8 is 0.9905121844144327
Reliability for VM = 9 is 0.988144632660903
Reliability for VM = 10 is 0.9920372733294893
```

Fig. 5: Reliability framework using fault tolerance technique

In the above framework [Fig. 5], fine-grained check pointing is used to increase the reliability.

CONCLUSION

As we know that failures are inevitable in the cloud environment. To solve this problem, in this paper an adaptive fault tolerance method is proposed to provide reliability of cloud nodes. To handle the major challenge of tolerating fault related complexity, a study was done over cloud computing and introduced fault reduction scheme. In pre-emptive cases, the requests from users were influenced by this scheme for scheduling VMs. Also increment in using adaptive fault tolerance techniques provides reliability to the node outline. The reliability can also be increased by using fine-grained check pointing method which lowers node failures. The response time and availability is taken into account for performance enhancement. Backward recovery technique is be implemented to make the system fault tolerant. The adaptive nature of the proposed method specifies that the cloud's performance will be appreciably enhanced. The response time and availability is taken into account for performance evaluation. The reliability can also be increased by using replication method which can be implemented in upcoming research papers.

CONFLICT OF INTEREST

There is no conflict of interest

ACKNOWLEDGEMENTS

None

FINANCIAL DISCLOSURE

None

REFERENCES

[1] Dhingra M, Gupta G. [2017] Comparative analysis of fault tolerance models and their challenges in cloud computing. International Journal of Engineering and Technology, 6(2):36–40.

[2] Dhingra M, Gupta G. [2019] Architectural Framework for Cloud Reliability Model using Fault Tolerance Techniques. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(11):515-519.

[3] Mohammed A. [2016] Adaptive Framework for Reliable Cloud Computing Environment. IEEE Access.

[4] Limam S, Belalem G. [2014] A migration approach for fault tolerance in cloud computing. Int. J. Grid High Perform. Comput., 6(2):24–37.

[5] Cao J, Simonin M, Cooperman G, Morin C. [2015] Check pointing as a service in heterogeneous cloud environments Proc 15th IEEE/ACM Int. Symp Cluster, Cloud Grid Comput., Shenzhen, China, 61–70.

[6] Bobroff N, Kochut A, Beaty K. [2007] Dynamic Placement of Virtual Machines for Managing SLA Violations. Proc. of the 10th IFIP/IEEE International Symposium on Integrated Network Management, IM '07.

[7] Verrna A, Ahuja P, Neogi A. [2008] pMapper: Power and migration cost aware application placement in virtualized systems. Proc. of the 9th ACM/IFIP/USENIX International Middleware Conference.

[8] Sheheryar M, Fabrice H. [2011] Adaptive Fault Tolerance in Real Time Cloud Computing. IEEE World Congress on Services, 10:280-287.

[9] Meshram A, Sambare AS, Zade SD. [2013] Fault Tolerance Model for Reliable Cloud Computing. International Journal on Recent and Innovation Trends in Computing and Communication, 1(7):600-603.

[10] Damodhar M, Poojitha S. [2017] An Adaptive Fault Reduction Scheme to Provide Reliable Cloud Computing Environment. IOSR Journal of Computer Engineering (IOSR-JCE), 19(4): 64-73.