Fault Tolerance Detection, Mitigation and Performance Improvement in Cloud

S. Veerapandi, Dr. K. Alagarsamy
1Research Scholar, 2Associate Professor Rtd
School of Information Technology
Madurai Kamaraj University

Abstract: In the modern world computer become a necessary one. To work with computer system, software and hardware are necessary component. Due to technology development software and hardware prices has been increased nothing like anything. Purchasing new hardware & software become tedious process and expensive. Research found that cloud will overcome this kind of problem using the concept of usage based payment which growing as trendy technical solution and moving one. In cloud computing there are many problems persist but in our paper we have taken fault tolerance issue and sustainability. In this paper we have proposed a novel idea to improve the cloud fault tolerance and performance.

Keywords
Cloud Computing, MPI, Indexing Table

I. INTRODUCTION

Cloud based technology is sharing of resources on a larger scale which makes cost effective and location independent. Resources on the cloud can be uploaded by the vendor and used by the end user. Cloud also shares necessary software’s and hardware’s on-demand tools for various IT industries. Amazon has predicted the importance of Cloud computing at first then after Google and IBM. It is a model for enabling convenient, on-demand resource access to a shared pool of configurable computing resources like applications, networks, storage, servers, and services.

Cloud computing reached as a new computing methodology which provides reliable, and quality of service guaranteed computing lively environments for end-users. The growing demand for flexibility in obtaining and releasing computing resources in a cost-effective manner has resulted in a wide adoption of the Cloud computing.

Fault tolerance is a methodology which can continue to correctly perform their process in the presence of hardware failures, software or to operate satisfactorily in the presence of faults.

Thus, an important motivation for differentiating between faults and failures is the need to describe the fault tolerance of a system. An observer inspecting the internals of the system would say that the faulty component had failed, because the observer’s viewpoint is now at a lower level of detail.

II. REVIEW OF LITERATURE

Uesheng Tan, Dengliang Luo, and Jingyu Wang, “Cc-vit: Virtualization intrusion tolerance based on cloud computing,” [38] proposed a replication methodology for VMFT, exclusively managed by the cloud provider. It describes to improve efficiency by using passive VM replicas, which become active when a failure is detected. A methodology was introduced to transfer the state of VM. This solution is similar to the exclusive FT.

a collaborative solutions for specifics applications. However, they do not consider the splitting of the cloud between a provider and its customers, so their works are only applicable to an SaaS cloud.

OpenNebula, “Opennebula.org: The open source toolkit for cloud computing,” the Microsoft Windows Azure platform[33] offers an exclusive FT management at the cloud level. Windows Azure replicates each VM so that a VM failure is covered by its replicas. The Azure solution is limited to web applications developed in the Windows Azure platform. Moreover, no solution is proposed to repair the failed VM. In addition, for VMs which are not instantiated by the Azure development platform, the entire responsibility of FT management is left to the customer. This is also the case in the Amazon EC2 [34] cloud platform.

III. PROPOSED METHOD

After deep review of literature review we have found that main fault tolerance issue in cloud computing are recovery and detection. To solve these issues, many techniques were proposed by many authors but still fault tolerance is the major issue in cloud environment due to internet based, virtual mechanism, too many servers, lots scheduling, message passing system, etc. based on the study we have proposed a novel approach for detecting the fault tolerance and sustain in the fault tolerant environment. This is also very useful to reduce the future fault tolerance occurrence.

IV. PROPOSED ALGORITHM

1) Start
2) INT Task_CNT= 1;
3) Cloud System = True;
4) Index value = 0;
5) CNT = Task.CNT
6) If CNT > 1
7) Initiate job scheduler
8) Call MPI
9) If System = ok
10) Allocate Task
11) Update Index Table
12) Else
13) Not allot & Execute
14) Update Index Table
15) Continue until all task over
16) Else
17) No cloud Action

There are multiple tasks will run in the cloud environment, which controlled by job scheduler, which executes the tasks in cloud environment. Whenever scheduler executes the job in cloud computing, message passing interface sends the message to the system to check the availability of the system to execute the tasks. It will check the following attributes for the availability of the system.

- Speed
- RAM size
- Processor Bus
- Disk type
- Disk size
- Cache
- CPU Type
- SCSI Controller

While checking, there may be a chance to numerous system may not fit for executing which make system idle or non-suitable. In the existing method whenever task allotted scheduler will check every time to get the status of system, which makes the time consuming process and energy waste. To avoid this we have combined the concept of indexing table from the data structures. Indexing table will make the process simple, index node denote the availability of system and every sub table will contain the above parameters which are based on the previous tasks history. Index table will be automatically updated every time for every task response it will reduce the response time for new tasks. It was tested in amazon web services.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Existing Method (MPI) Response Time</th>
<th>Proposed Method Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0.438</td>
<td>0.281</td>
</tr>
<tr>
<td>14</td>
<td>0.527</td>
<td>0.306</td>
</tr>
<tr>
<td>21</td>
<td>0.605</td>
<td>0.425</td>
</tr>
<tr>
<td>28</td>
<td>0.623</td>
<td>0.467</td>
</tr>
<tr>
<td>35</td>
<td>0.681</td>
<td>0.498</td>
</tr>
</tbody>
</table>
V. CONCLUSION

There are numerous methods available for fault tolerance detection and task allocation. In this method we have proposed a novel approach based on message passing interface with indexing table for the improvement of fault tolerance detection and system availability check. We have simulated our idea with amazon web services and analyzed the performance of our algorithm which showed better response time and minimum checkpoints required compare with existing one. We believe that this research will be useful to modern technical society.

VI. FUTURE WORK

Every research has some benefit and loophole for future enhancement. Young research minds can take our research as benchmark and can develop some new fault tolerance sustainable mechanism using some new data structure technique. We hope that this research will be use full for young innovators.

REFERENCES

[6] Jean-clande Laprie “Dependable computing and fault tolerance: concepts and terminology” LAAS-CNRS 7 Avenue du Colonel Roche, 31400 Toulouse, France


