IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol-6, Special Issue-3, Apr 2018 E-ISSN: 2347-2693

Study of Various Proactive Fault Tolerance Techniques in Cloud Computing

Atul Kumar[1*], Deepti Malhotra[2]

[1*]Department of CS & IT, Central University of Jammu, Samba, India
[2] Department of CS & IT, Central University of Jammu, Samba, India

[*]Corresponding Author: kumaraatul1993@gmail.com, Tel.: 7006855787

Available online at: www.ijcseonline.org

Abstract— Cloud computing works in a distributed environment that is very complex in nature. Due to its large size and high complexity, emergence of fault can happen anywhere and anytime. Since cloud is a distributed platform, so it is more prone to errors and failures. In such an environment, avoiding a failure is difficult and identifying the source of failure is also very complex. Fault in cloud may lead to loss of critical data or trust loss of the customer in the organization. Data failure may be due to some corrupted data, missing source file or some other flaw in the data. Different fault tolerance techniques are used to achieve the robustness of the system. These techniques maintain the robustness and dependability of the cloud network. Many Researchers have employed the different techniques for self healing, aging detection and rejuvenation of virtual machine. This research paper presents the various proactive fault tolerance mechanisms that reduces the consumption of computing resources and enhances service availability to large extent.

Keywords— *cloud computing, self healing, aging detection, rejuvenation*

I. INTRODUCTION

Cloud computing is a pool of resources that are utilized on-demand as a service by user in a distributed environment. In cloud, user is not worried about the maintenance of system or software. That is done by cloud providers themselves. Cloud can be considered as a reliable platform because of its features like heterogeneous, distributed and totally virtualized resources, dynamically provisioning of resources, pay-as-you-go model etc. Cloud is broadly classified into three categories: *IaaS* : various resources are rented like storage, servers, networking and data center space as pay per-use basis to companies. *PaaS* : provides a complete lifecycle to build and deliver web based applications like developing, testing, delivering managing. Customers do not have to worry about setting up and managing of resources. *SaaS*: it is a method for delivering software applications over the internet. Cloud providers host and manage the software application and Handle any maintenance like upgrade security patches etc. In general cloud computing infrastructure are built by interconnecting large datacenters and resource are available via internet to user in form of on-demand service. Due to large size and complex nature, cloud is vulnerable to fault and failures. This may lead to hazardous loss to the service providers as well as users. So there are lots of techniques used for making the system faultless either by removing fault or by avoiding it. When some fault occurs in the system there should be such techniques that can handle or take appropriate action against that fault. These techniques are generally categorized into: Proactive Fault tolerance & Reactive Fault Tolerance. [1][2]

In this paper proactive fault tolerance and its types are discussed. Sections 2 of the paper discuss proactive fault tolerance technique and its types in detail. In section 3 comparisons of various feedback loop control mechanisms is done .Some terminologies that are used in the paper are discussed in section 4 and in section 5 conclusion is given.

II. PROACTIVE FAULT TOLERANCE IN CLOUD COMPUTING

The principle of proactive fault tolerance policies is to avoid extra effort for recovering the failed job, nodes, by predicting failures in advance and proactively replace the suspected components with other working components. Proactive fault tolerance systems are able to fulfill the time constraints set by the real time systems. It can be categorized as: Software Rejuvenation**,** Proactive Fault Tolerance using Self-Healing& Proactive Fault Tolerance using Preemptive Migration. [3]

A. *Software rejuvenation:* Software rejuvenation is a technique used to control system performance degradation and software aging related failures. The mechanism involves different steps such as stopping application periodically and restart after clearing the existing internal state. Software rejuvenation performs Clearance of garbage, buffer queues flushing, and internal kernel tables reintialization and file systems cleaning up. Intrinsically, operating environment of application is cleaned and restored by it. Rejuvenation process, methods may vary according to the approach they are applied. Some techniques like for pre-determined time interval, time based methods are used for rejuvenating the system. The other techniques in practice are inspection-based which involves the monitoring of aging indicator metrics and apply rejuvenation mechanism during the forecasted period. Some other techniques used are inspection-based that involves the monitoring of indicator metrics of aging and applying rejuvenation process during the forecast period. The service may get disrupted due to use of reboot mechanism in rejuvenation process that may affect the business also. Important issue for research is to find when and how many times the software must be rejuvenated and the techniques to be used to avoid the aging effects of software. Researchers have used various techniques for detection of aging and rejuvenation.

- *Software Aging Detection:* Software aging means software performance or functionality degradation caused by exhaustion of resources mainly due to memory-leaks, unreleased locks, non-terminated threads, or storage fragmentation. Predicting exhaustion time of resources is necessary to avoid system crashing. In case of rejuvenation, monitoring of multiple metrics is necessary like in case of VMMs that reflect broader resource utilization. The metrics like CPU usage, memory usage, and network bandwidth and disk access data can be used, as the aging directly impacts the resource performance. Data is collected and analyzed to indicate any trend related to software aging pattern is detected due to resource exhaustion.
- *VM Migration:* After detection of software aging, rejuvenation process is applied on aged VMMs which can be done by rebooting. But, before the reboot, migration of all VMs must be done to some healthy VMM. Live migration of VMs is the technique that migrate Virtual Machine from one physical machine to another physical machine without causing outages.[4]

Clouds are a large pool of resources .Because of it large size and high complexity, there is need for self healing too. Self-healing can to be implemented via three necessary components, when consequence oriented concept is considered i.e.: Prescriptions, Healing Categories, and Recursive Healing Method. Different requirements are fulfilled, aimed for different prescriptions accordingly to recover or to prevent serious consequences that make possible certain prescription by sacrificing others essentially in case of some conflicts. Some requirements are in conflict or present tradeoffs. To examine threatened levels, corresponding categories will be chosen from various self-healing approaches. Some self healing categories are:

i. *Minor severity level*: Possible consequences are diagnosed while patient process is running that decides the corresponding prescriptions for healing it.
ii. *Major severity level*: To detect the consequences and to prevent or recover from the consequence, diagnosis starts immediately by suspending the patient process.
iii. *Serious level*: The fastest and easiest way to handle the potential patient process that can cause destroy the entire system is to kill that dangerous process. However, the host should have start another backup process to run for the same function, before it becomes a serious threat to the system.
iv. *Catastrophic level*: Reboot the system if catastrophic failure is detected by host. System can be restarted from the latest checkpoint i.e. created by monitoring and storing states of all the processes, after rebooting is done.

Apart from above mentioned four different categories for healing, more categories can be added according to the severity levels that will not affect the generality of the framework. The healing process starts to recover from the possible consequence also called *diseases hereafter* after identification. The recursive healing method is presented as follows:

The host determines and runs a prescription after diagnosis for healing of identified disease. After healing, results are fed back. However, if the problem still exists, then the second option can be tried for output of the prescription by hybrid diagnosis or another

B. *Proactive fault tolerance using self healing:* diognosis is done and some other prescription is applied. These steps can be recursively attempted to mitigate the consequence for several rounds. If all attempts are found to be worthless then; the corresponding processes can be restarted by the host or reboot operation is performed .Some important variables or states can be recorded before the restart operation of the system in order to return to the last healthy check-point. For learning purpose, the healing result will be delivered to the diagnosis modules. As there is a chance of insufficient resources required for healing. So in that case the patient host may be assigned to some other host by the system for healing. The recursive method not only applies multiple prescription for different type of possible consequences assigned by the hybrid diagnosis but also uses other host to realize peer healing if the current host lack resources for healing purpose .[5]

C. *Proactive fault tolerance using preemptive migration*: Proactive FT using preemptive migration based on a feedback-loop control mechanism (Figure 1), where each applications health is monitored which is running on a High-Performance Computing (HPC) system. Also parts of running application are reallocated to fit nodes from unfit one to avoid any kind of upcoming failure. Continuous monitoring of application health, in presence of threads reallocating application parts and application allocation reflection in application health forms the feedback loop. Both software and hardware are considered to monitor application health, like in case of hardware observing speeds of fan, processors temperature, and processors utilization. Whereas in case of software it may even include applications progress watching, such as I/O patterns, similar to performance monitoring. Filters may be used to process the monitored data and an online reliability analysis to identify trends, patterns, correlations, imminent failure indication, and future possible threads. The feedback loop control mechanism may consider performance parameters, such that to evaluate system and application health in terms of *performability.*

Reallocation evicts parts of application from one or more nodes and removes them from further use. Migration is either performed to currently unused nodes, already allocated ones (oversubscription) or reserved one, due to reduction of pool size of nodes. Excluded nodes are manually inspected by the system administrator before adding them to the pool e.g., in case of a fault in fan, and/or an automated testing procedure, e.g., in case of unusually low processor utilization. Reallocating parts of an application typically involves the system resource manager and the compute node runtime environment.. [6]

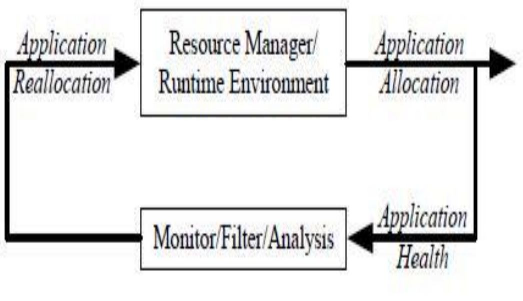

Figure 1: Feedback-loop control mechanism using preemptive migration

Classification:

Four distinct types of proactive FT based on loop feedback mechanism design using preemptive migration can be derived as:

i) Type 1

Type 1 is the most basic type depicted in figure 2. System and application health parameter, like processors temperature, fan speeds, and processor utilization are observed by monitor present on each compute node. The Monitor detect alerts and exceeding preset limits like fan faults, increase in temperature of processors etc and notifies Resource Manager to evict application from its compute node. The Resource Manager reallocates the parts of application and notifies Runtime Environment to migrate parts of application to other nodes. The mechanism covers basic failure types like fan fault that may lead to overheating of processor and result in shutdown of system automatically. This cannot be evaluate the history and also very vulnerable to false positives, i.e. migrating when not required and migrating when not required i.e. false negative. This type also misses the real time window in which migration is needed to be done to avoid upcoming failure.

ii) Type 2

Type 2 shown in Figure 3 is extended version of the Type 1 that offers a better approach for basic failures. Using short term historical context, sensor data is processed with the help of Filter. This results to clear picture for identifying imminent failures and through trend analysis the future possible threads. This has less false negatives and false positive than Type 1 and also don’t miss the real time window.

iii) *Type 3*

The type 3 is the upgraded version of type 2 as revealed in figure 4. Reliability of each system and its running application is modeled by additional feature of type 3 known as reliability analysis. Type 3 has facility to analyze application health status that is not present in case of type 1 and type 2.Also applications health is not degraded in this due to migration. This type also has less false positive and false negatives increases its Quality of Service than type 1 and type 2.

iv) *Type 4*

The Reliability Analysis of Type 4(Figure 5), uses a History Database and application reliability patterns. So that it can match them with the current pattern being experienced to optimize the mechanism of feedback loop control system through various techniques of machine learning. The Reliability Analysis may make use of Database of history to perform offline system's investigations and reliability patterns of application, when long term records and more computation is required. The QOS is Higher than all above mentioned.

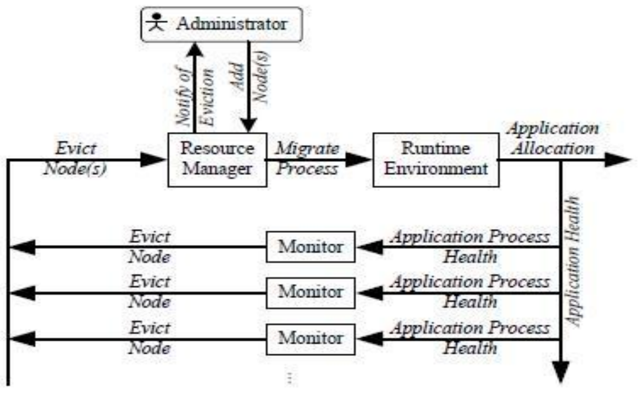

Figure 2: Type 1 of feedback loop control mechanism

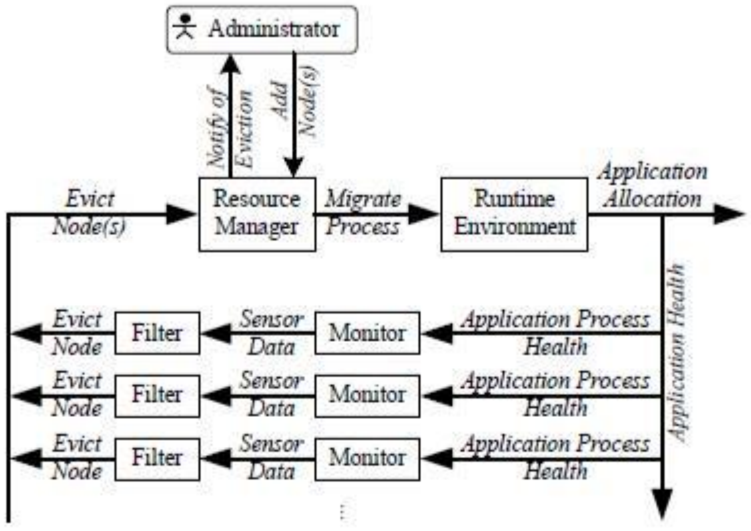

Figure 3: Type 2 of feedback loop control mechanism

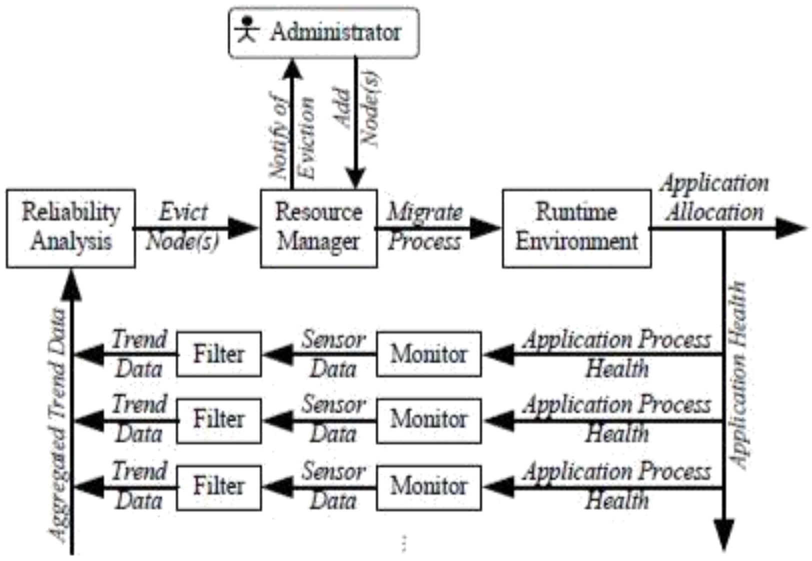

Figure 4: Type 3 of feedback loop control mechanism

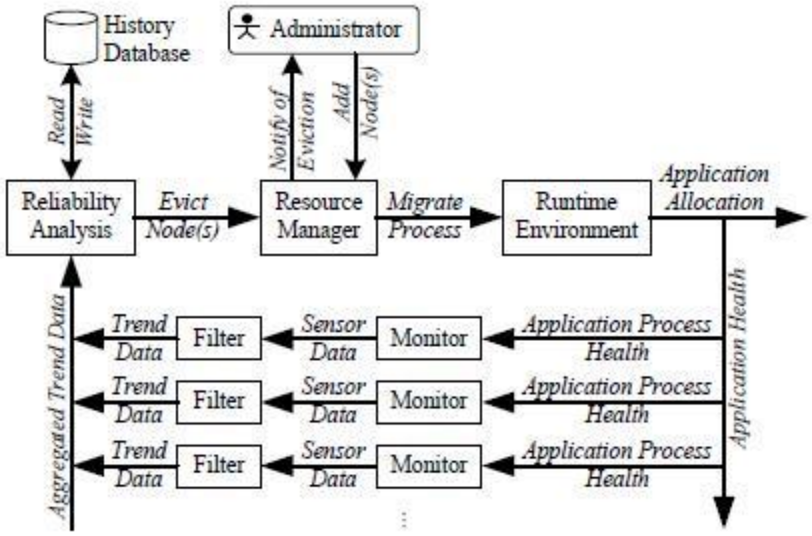

Figure 5: Type 4 of feedback loop control mechanism

III. COMPARISON OF VARIOUS TYPES OF FEEDBACK LOOP CONTROL MECHANISM

Table 1: Comparison between various feedback loop control mechanism types in Table 1

Types	**Type 1(a1)**	**Type 2(a2)**	**Type 3(a3)**	**Type 4(a4)**
Quality of Service	good	Better than a1	Better than a1 & a2	Better than a1,a2 & a3
Application health Status view	No	No	Yes	Yes
Offline investigation of Systems	No	No	No	Yes
Real-time Window	Prone to miss	Prone to miss	Able to maintain	Able to maintain
Prone to false positive/false negative	high	less	less	less

IV. TERMINOLOGIES USED

- *High Performance Computing:* HPC is the use of “parallel processing” techniques to solve complex computational problems through computer modeling, simulation and analysis. Most of the users of HPC systems have been academic institutions scientific researchers, engineers and specific government agencies like in military. Cloud properties like scalability, availability on demand and quick to deploy had encourage businesses to consign assets expenditure for HPC infrastructure.[7]
- *VMM:* A Virtual machine screen (VMM) is a software system that empowers the creation, management and governance of virtual machines (VM). Furthermore, it manages the operation of a virtualized domain with respect to highest priority on a physical host machine. VMM is also known as Virtual Machine Manager and Hypervisor. [8]
- *Performabilty:* Cloud is evaluated on the basis of performance of its services. Evaluation can be done on various criteria like average waiting time, throughput ,load balancing response time ,release of resources etc.[9]
- *Oversubscription:* It means offering resource more than its actual capacity by assuming that most of the users will not use consume the entire portion of their resource .Main motive is to reduce wastage and increase the profit. [10]
- *Quality of service:* QOS can be considered as the certain degree of performance that is guaranteed by the cloud provider to the user. [11]

V. CONCLUSION

Fault tolerance techniques are necessary part for cloud. As data stored on cloud is vulnerable to various faults and failure because of many reasons. So some measure must be taken to ensure such data is not lost .So a proactive approach discussed above that will predict the occurrence of failure earlier, instead of acting after occurrence. That is why it is considered better approach than reactive techniques that is applied after faults occurs. Although various techniques are mentioned above but still lot of work is needed to be done for improvement in Proactive technique to make system more fault tolerant.

References

[1] Soma Pratibha , S. Sowvarnica ,” *Survey of Failures and Fault Tolerance in Cloud*”,Second International Conference On Computing and Communications Technologies (ICCCT’**17**), **2017**.

[2] Youssef M.Essa ,” *A Survey of Cloud Computing Fault Tolerance: Techniques and Implementation*”, International Journal of Computer Applications (**0975 – 8887**), Volume **138** – No.**13**, March **2016**.

[3] Harpreet kaur, Amritpalkaur,” *A survey on Fault tolerance techniques in Cloud Computing*”, International Journal of Science, Engineering and Technology, April **20, 2015**.

[4] I.M Umesh, Dr. G N Srinivasan, Matheus Torquato,"*Software Rejuvenation Model for Cloud Computing Platform*" International Journal of Applied Engineering Research, Volume **12**, Number **19** (**2017**) ,pp. **8332-8337**.

[5] Yuanshun Dai, Yanping Xiang, and Gewei Zhang, " *Self-healing and Hybrid Diagnosis in Cloud Computing*", CloudCom **2009**, LNCS **5931**, pp. **45–56**, **2009**.

[6] Engelmann, G. R. Vallee, T. Naughton, and S. L.Scott, "*Proactive fault tolerance using preemptive migration*", Euromicro International Conference on Parallel, Distributed, and network-based Processing (PDP), pages **252–257**, **2009**.

[7] Avinash Nidumbur, "*high-performance computing*", [online]available: https://www.ibm.com/blogs/cloudcomputing/**2016**/**06**/high-performance-computing-cloud/

[8] virtual machine monitor[online] , available : https://www.techopedia.com/definition/717/virtual-machine-monitor-vmm

[9] Niloofar Khanghahi and Reza Ravanmehr," *CloudComputing Performance Evaluation: issues and challenges*", International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.**3**, No.**5**, October **2013**

[10] Rachel Householder, Scott Arnold, Robert Green," *On Cloud-based Oversubscription*", International Journal of Engineering Trends and Technology (IJETT) – Volume **8** Number **8**- Feb **2014**.

[11] Danilo Ardagna, Giuliano Casale, Michele Ciavotta, Juan F Pérez and Weikun Wang," *Quality-of-service in cloud computing: modeling techniques and their applications* ", Journal of Internet Services and Applications **2014**

Authors Profile

Mr.Atul Kumar pursed Bachelor of Engineering from Model Institute of Engineering & Technolgy, J&K in 2015 and currently pursuing Master of Technolgy from Central University of Jammu, J&K, India, 2018. His current research interest includes reliability and availability by means of fault tolerance in Cloud Computing and Internet of Things.

Mrs.Deepti Malhotra pursed Bachelor of Engineering in Computer Science from Model Instiute of Engineering & Technolgy, Jammu, India in 2005 and Master of Engineering from Thapar University, Punjab, India in 2007 in Computer Science. She completed her Ph.D in Computer Science from University of Jammu, India in 2013 and currently working as Assistant Professor in Department of CS & IT, Central University of Jammu, India. She is Lifetime Member of Indian Science Congress Association and member of the Institution of Engineers (India).She has published many Research papers in international Journals and Conferences .Her area of interest include Parallel & Distributed Computing ,Grid Computing, Cloud Computing and Data Mining

.