Volume 8, No. 3, March – April 2017
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Fault Prediction and Mitigation in Cloud Computing

Vidhi Sutaria
Computer science and Engineering
Institute of Technology
Nirma University
Ahmedabad, Gujarat, India

Vivek K Prasad
Computer science and Engineering
Institute of Technology
Nirma University
Ahmedabad, Gujarat, India

Dr. Madhuri Bhavsar
Information and Technology
Institute of Technology
Nirma University
Ahmedabad, Gujarat, India

Abstract: Cloud computing provides on-demand service in computing technology. This technology offers integration of soft-ware and resources which exhibits dynamic scalability in nature. These systems are beneficial, but there is also disadvantage for the same. If the system has symptoms of failure, then it does not provide its functionality. So the solution is fault prediction and mitigation. It provides the capacity to the system to react smoothly when the system has gone wrong or any unexpected hardware or software failure in the organization. This paper illustrates a better reason of defect prediction and mitigation methods, techniques and tools used for fault prediction and mitigation in cloud computing. In this paper, the techniques of fault Prediction and mitigation has been discussed and implemented.

Keywords: Cloud Computing, Fault Prediction, Fault Mitigation, Proactive, Reactive, Hardware Fault, Software Fault.

I. INTRODUCTION

A. Cloud Computing

Cloud computing is an innovation in range of virtualization, network computing, utility registering. Cloud computing provides IAAS (Infrastructure As A Service), PAAS (Platform As A Service) and SAAS (Software As A Service)[1]. IAAS provides service to network architects. PAAS provides service to application developers. SAAS pro-vides service to end users [1].

B. Fault Prediction and Mitigation

Fault Tolerance is the procedure of finding faults and failures in a system. If a fault occurs or there is a hardware failure or software failure, then also the system should function properly. Failures should be handled in a dynamic way for reliable Cloud Computing. It will also give availability and robustness against the system hardware and software failure into the organization.

C. Benefits to implemented Fault Prediction and mitigation system

Fault prediction and mitigation has two types of techniques: proactive and reactive. The Proactive fault tolerance policy is to avoid recovery from failure by predicting them and proactively replace the suspected component means detect the problem before it actually come. Reactive fault tolerance reduce the effect of failures on application execution when the failure effectively occurs. The effect of failure is reduced when the failure actually occurs[1].

II. TYPES OF FAULT

The inability of a system to perform its required task caused by an anomalous state or a bug in one or more than one part of a system. Faults are the hypothesized or adjudged cause of an error the main cause which causes an erroneous belief [2].

Hardware Fault:

Most of the fault-tolerant strategies have focused towards structuring systems that can recover themselves from the faults that usually happen in hardware modules, this involves splitting a computing system into modules. So if a particular module gets filled, another module can keep on doing its functionality [3].

Software Fault:

It is similar to hardware approach but here more con-sideration is on tolerating faults at the software level [4]. For achieving this various static and dynamic redundancy approaches has been used [3].

A. Fault Prediction and Mitigation Techniques in Cloud Computing

Proactive Fault Tolerance:

This method of is to avoid extra effort for recovering the failed tasks, nodes, by predicting the fault in before and replace them with other working parts. Proactive fault tolerance systems are able to fulfill the time constraints set by the real time systems [5].

Reactive Fault Tolerance:
When an actual fault occurs in the system, then this method help to reduce the impact of failures on the running system. These techniques provide good fault tolerant solution for gen-eral computing application, but a problem with this technique is that it cannot fulfill the time constraints set by real time computing systems [5].

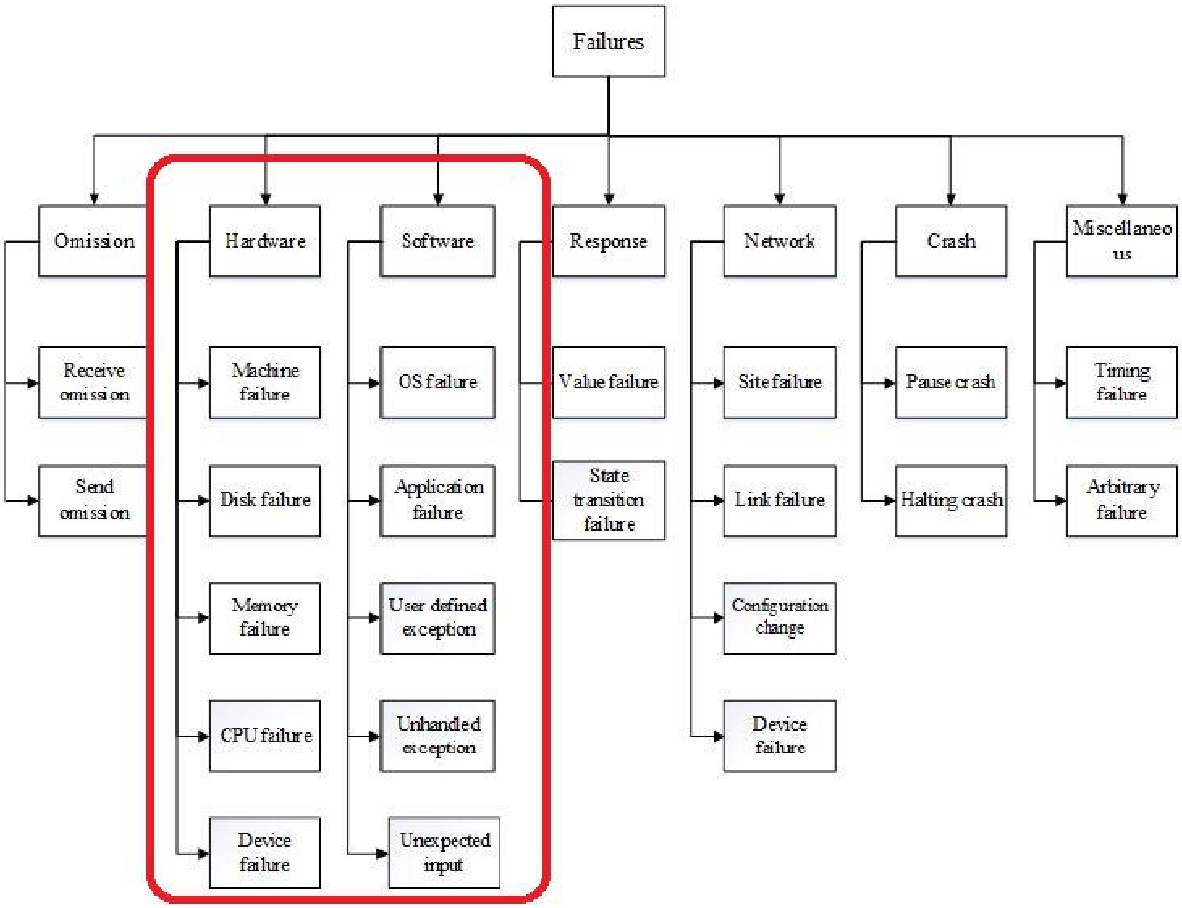

Fig. 1: Types of fault[2]

B. Types of Proactive Fault Tolerance

Software Rejuvenation:
Programming Rejuvenation is the technique in which an application is promptly ended and afterward restarted as a spotless state. There are different restoration interim and the application is restarted at each interim with a clean inward state. In this method, intermittent reboots are planned for the framework [6].

Using Self-Healing:
When multiple components of a single system are running on different multiple VM and when a fault occurs by using self-healing the failure of different application instances can be handled automatically [6].

Using Preemptive Migration:
By using this technique, the parts of a system running on a computing node that is about to fail are migrated to different nodes. By using preemptive migration the application is migrated to a different node before actual fault occurs [6].

C. Types of Reactive Fault Tolerance

Check pointing/ Restart:
This method used when doing task scheduling, the check-points are inserted to identify fault incidence. These techniques take less computation and less time as a result of the task is restarted at the previously checked point. There is no ought to restart the full task [6].

Replication:
Replication is a very effective technique in fault tolerance. In this, there are various replicas of any application or task on different resources. When any fault occurs in the system then execution continues to succeed until all replicated tasks are destroyed [6].

Job Migration:
When due to resource failure or machine failure any task fails then the task is shifted on another virtual machine. Where it continues its execution of process [6].

Sguard:

It requires less registering investment to ordinary process time and accessible more assets free. It is depended on rollback-recovery. It takes less time to normal process and makes a lot of resources out there [6].

Retry:
In this method first of all the unsuccessful task is re-executed on the same resource from starting. If the task continues to fail on different executing points, then to get rid of the additional overhead the average execution time is computed. A threshold value is set to limit the number of retries of the failed job on the same machine [6].

Task Resubmission:
Task resubmission means that when any task fails then it is recommitted either to the same machine or a different one.[6].
User defined exception handling:
When any task/job fails then the user gives the efficient treatment to that failed jobs for work flow. In exception, handling user can write code into the try, catch and finally block to throws exception and it handles the exception [6].

Rescue workflow:
This technique permits the work stream to proceed regardless of the possibility that the undertaking comes up short until it gets to be difficult to push ahead without cooking the fizzled task. [6].

III. MODELS AND TECHNIQUES OF FAULT PREDICTION AND MITIGATION

A. Comparison among various model

Table 1 described comparison among various model. AF-TRC (Adaptive Fault Tolerance) a proposed distributed computing. Which is based on constant framework and it gives advantage as the figuring limit, adaptable, and virtualization for a continuous framework. LLFT (Low Latency Fault Tolerance) is a given model which conveys a low dormancy, adaptation to non-critical failure (LLFT) middleware for giving adaptation to internal failure to dispersed applications sent inside the distributed computing environment as an administration offered by the proprietors of the cloud. FTWS is a given model which gives a replication and resubmission and assemble work process planning calculation for giving adaptation to non-critical failure by utilizing the need of the assignments. FTM is a model which is conquer the constraint of existing strategies for the on-request service [7]. The Vega-superintendent is work for virtual group to beat the 2 issues: ease of use and security. Magi-Cube a very tried and true and low excess stockpiling design for distributed computing [8][9].

B. Various Fault Prediction and Mitigation Techniques

Fault tolerance challenges and techniques have been implemented using various tools. Table 2 compares these tools based on their programming framework, environment and application type along with Different fault tolerance techniques.

TABLE I: Various Fault Tolerance Models [10]

Model Name	Protection against type of fault	Applied procedure for tolerance the fault
AFTRC	Reliability	1. Delete node, depending on Their reliability 2. Back word recovery with the help of check pointing
LLFT	Crash-cost, trim-Ming	Fault Replication.
FTWS	Dead line of work flow	Replication and resubmission of jobs
FTM	Reliability, availability, on demand inspection and repair	Replication user's application and in the case of replica failure use algorithm like gossip based Protocol.
CANDY	Availability	1. It assembles the model com-ponents generated from IBD and STM according to allocated-Then activity SNR is synchronized to system SRN by identifying the rela-tionship between action in ac-tivity SNR and state transition in system SRN.
VEGA-WARDEN	Usability, security, scaling	Two authentication layer and standard technical solution for the application.
FT-CLOUD	Reliability, crash and value fault	1. Significant component is de-fined based on the ranking. 2. Optimal ft technique is deter-mined.
MAGI-CUBE	Performance, re-liability, low stor-age cost	1. Source file is encoded in then splits to deliver as a cluster. 2. The file recovery procedure is triggered is the original file is lost.

Disappointments Hadoop is utilized for information escalated applications, yet can in like manner be utilized to actualize adaptation to internal failure systems in a cloud situation. Amazon Elastic Compute Cloud (EC2) gives a virtual registering environment to run Linux-based applications for adaptation to internal failure [5].

IV. RELATED WORK

Deepali Mittal, and Ms. Neha Agarwal designed model on Fault Tolerance in Cloud Computing [12]. In this described Fault Tolerance is the methodology used here is finding faults and failures in a system. In this proposed Dependability Assessment Algorithm and Decision Mechanism Algorithm were discussed [12].

Pankaj Deep Kaur and Kanu Priya defined Fault Tolerance Techniques and Architectures for fault tolerance [3]. It presents, in brief, the need and matrices for performing fault prediction and mitigation in the cloud. It gives an outline of the prevalent architectures and the existing techniques for that have been analyzed and compared[3]. Himanshu Agarwal and Anju Sharma have proposed Fault Tolerance Techniques in Cloud Computing [2].

TABLE II: Tools Used To Implement Existing Fault Tolerance Techniques

Fault Tolerance Techniques	System	Programming Framework	Environment	Fault Detected	Application Type
Self Healing, Job Migration, Replication	HAProxy	Java	Virtual Machine	Process/node	Load balancing Fault Tolerance
Chek-pointing	SHelp	SQL,JAVA	Virtual Machine	Application Failure	Fault tolerance
Check pointing, Retry, Self Healing	Assure	JAVA	Virtual Machine	Host, Network Failure	Fault tolerance
Job Migration,Replication, Sguard,Resc	Hadoop	Java, HTML, CSS	Cloud Environment	Application / node failures	Data intensive
Replication, Sguard,Task Resubmission	Amazon EC2	Amazon Machine Image, Amazon Map	Cloud Environment	Application / node failures	Load balancing, fault tolerance

Prasenjit Kumar Patra and Harshpreet Singh introduced fault Tolerance Techniques and Comparative Implementation in Cloud Computing [1]. It has been classifying that to give a superior comprehension of blame recoup methods utilized for as a part of cloud situations with some effectively characterized model and look at them [1].

Harpreet Kaur and Amritpal Kaur described a Survey on fault tolerance techniques in cloud computing [13]. This is described to study different types of failure and different techniques for handling them. By taking a correct action before or after the failure arrived, the system can be predicate fault within it and recover the fault [13].

Priority Scheduling Algorithm with Fault Tolerance in Cloud Computing introduced by Seema Bawa and Nimisha Singla[15]. In this, it portrays new need booking calculation with adaptation to non-critical failure.

V. PROPOSED SYSTEM DESIGN

For fault prediction and mitigation we proposed system design. Which is described in figure 2 and figure 3 for fault prediction and mitigation in the system. Which first of all identify the fault and predict the fault using Fault Tree Analysis (FTA). According to that, it will identify by the error code if it is hardware fault or software fault. After identifying the reason behind the system failed it will mitigate the fault to recover the system fault. So that the system take action according to the Event Tree Analysis(ETA). So, by collaborate FTA with ETA it will predict the system fault, identify the reason behind it and recover the system fault.

In given figures, it describes various techniques of fault prediction and mitigation. For fault prediction, it defined various faults among them one of the fault can be predicted by the fault tree analysis. For fault mitigation arrival fault can be mitigate by the particular mitigation technique. By this way, the system can predict the fault using fault tree analysis and mitigate these fault using event tree analysis.

A. Software failures codes

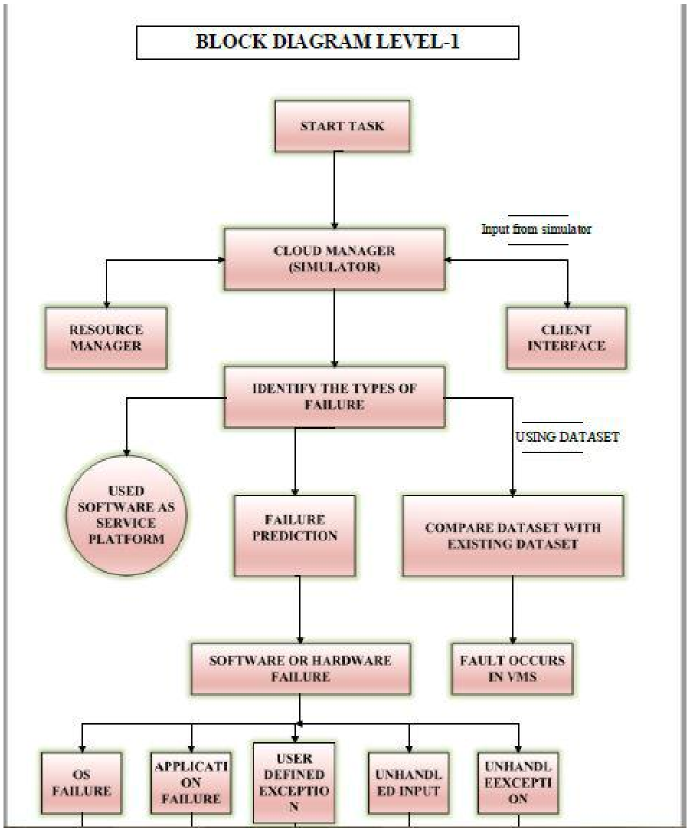

Fig. 2: System Design For Fault Prediction

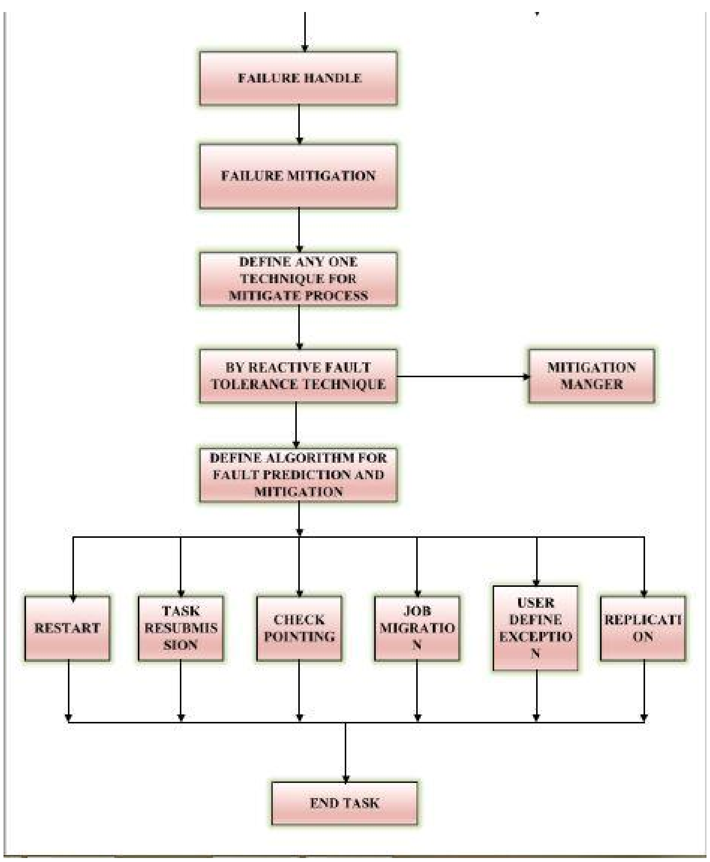

Fig. 3: System Design For Fault mitigation

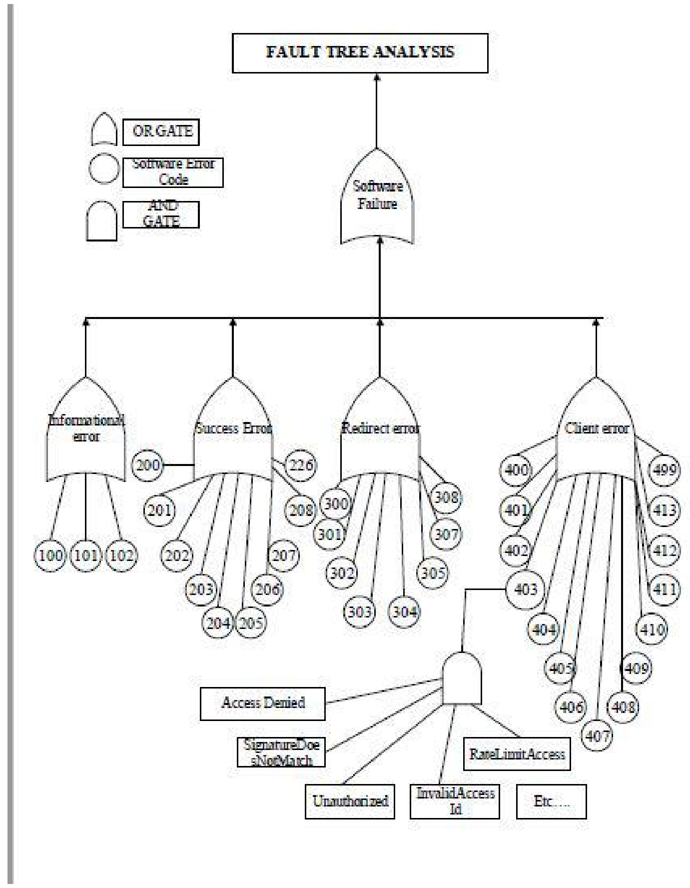

Fig. 4: Software Failure codes

B. Hardware failures codes

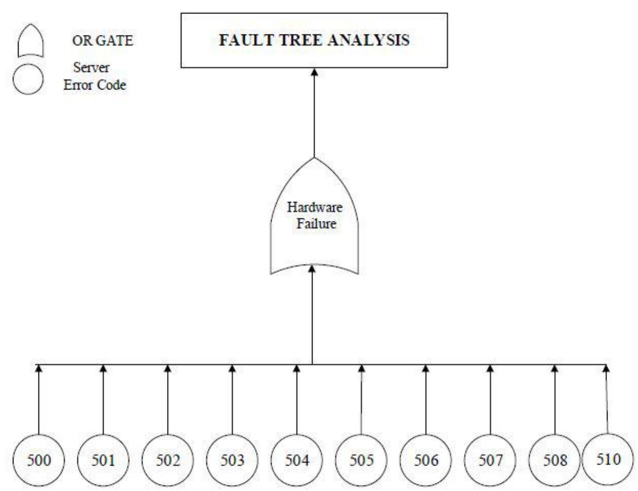

Fig. 5: Hardware Failure codes

In the given figure 4 and figure 5, it describes hardware and software failure cloud error codes. By which, we can identify the particular one failure and reason behind it.By that code, we can predict the particular failure into the system and take action according to that fault. There is one particular cloud error code for each and every error if it is hardware error or it is a software error. By this cloud error code, we can identify the root cause .of failure. By this, we can identify the error and reason behind it.

C. Proposed Working Model

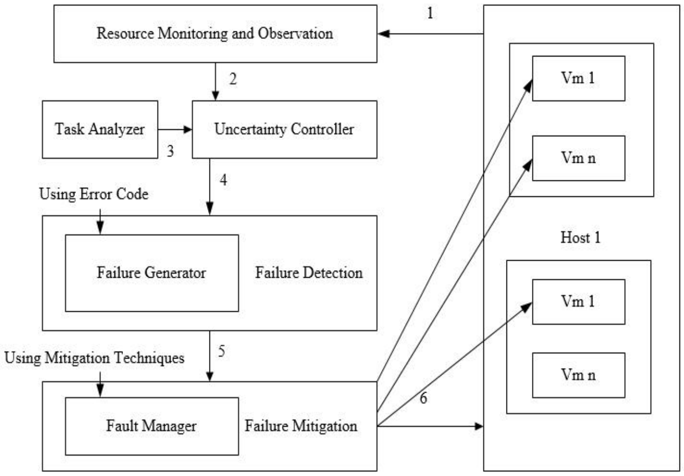

Fig. 6: Working Model

In the given figure 6 it describes proposed working model, described the concept fault prediction and mitigation model will be run on the cloud environment. As given figure shows assuming that, number of virtual machines listed on one host. This is known as one cloud. Resource monitoring and observation module runs on cloud which is provides the resource monitoring and observation of the cloud then any fault occurs in the cloud which is observed by the uncertainty controller. Then after faulty task maintain into failure detection module which is provided the prediction of the fault by the fault error code. Various error codes are describes in the above figure. Various hardware and software error code are specifically defined for the particular error. After fault prediction fault mitigate by the failure mitigation module. In this failure mitigate by the various fault mitigation techniques. Various fault mitigation techniques are defined in two techniques. Such as proactive mitigation techniques and reactive mitigation techniques. For this scenario used reactive techniques such as restart, task submission, job migration, and user defined exception handling, replication, and rescue workflow. For every error code specific mitigation technique are defined. Which is defined into the below table 3. Mitigation techniques for error code.

So, By that way this model describes how to predict error and mitigate the error in when it is occur into cloud.

VI. PROPOSED ALGORITHM

Algorithm 1 Initialization

```
Let the current load is assumed
{
Initialize the task scheduling for every server s[i]
When new task arrives check load server ().
if (load server s[i] < 15) then
    {
    Skip task to next server
    Check t[i], Sh[i].
    put the task in waiting list ;
    redirect to fault tolerance();
    Minloadserver();
    }
else
    (load server s[i]>15)
    {
    Check the task status;
    Redirect to fault tolerance();
    Minloadserver();
    }
end if
```

Algorithm 2 Fault prediction and mitigation algorithm

```
if( status= "W") then
    {
    Scheduled the task to that server;
    Go to next server status ;
    }
else (status= " F" )
    {
    Check type of fault
    if (shf[i]==vmf[i]) then
        {
        Check the status code;
        if (100 || 101 || 102 ) then
            {
            Informational error;
            }
        end if
    else
        if (200|| 201 upto || 204) then
            {
            Success error;
            }
        end if
    else
        if (302 ||303 upto || 308 ) then
            {
            Redirect error;
            }
        end if
    else
        if (400 || 401 upto || 499) then
            {
            Client error;
            }
        end if
    end if
end if
if (500 ||501 upto || 510) then
    {
    Server error;
```



```
    }
end if
}
Find minloadserver s[j]
{
sh[j]<=5(Minimum Load)
(5<=sh[j]<= 10(Average Load))
(10<=sh[j]<=15(Heavy Load))
}
The s[j]'s new sh[j]=sh[j]+shf[i]
{
Return to the next task;
}
end task;
}
```

This algorithm will plan the undertakings introduce on every server at some moment parameter and after that it will additionally continue to next server when an option to the new assignments by its needs and it is done with FCFS (First Come First Serve) algorithm. This calculation principally concentrates on the blame tolerant nature of the calculation to handle the blunder. It takes situation when the bit of assignments running on every server is same.

VII. RESULT

Fault prediction and mitigation implemented on Environment: JAVA, IDE: Eclipse, Simulator: WorkFlowSim-1.0. For fault prediction it is use decision tree method in data mining and for this it is implemented in R statistical tool. By decision tree method, results are as follows which decries in below figure 7.

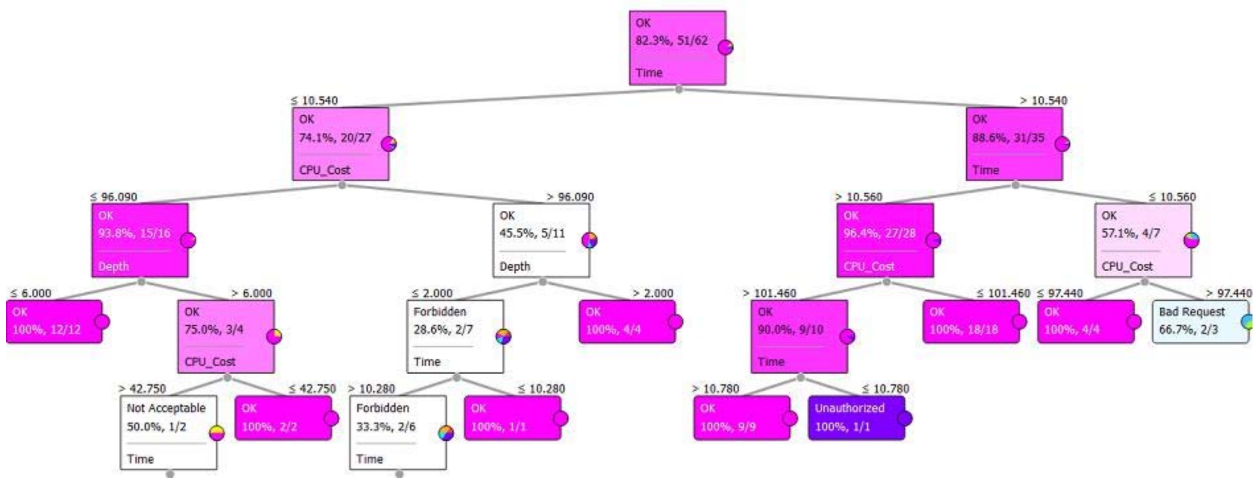

Fig. 7: Fault Tree

As figure 7 shown, by prediction result using decision tree we can do analysis of different error code. Various errors can be assorted based on the performance parameters such as execution time of the task, CPU cost, RAM cost, Depth of the task. By this parameter which error can be arrived that can be augured by the decision tree. It can be predicted which error will be come next by its percentage ratio. By decision tree it classified the task will become success or failed. If the task will be failed then which error will be occurs. Each error can be mitigate by the specific mitigation technique.

By fault prediction and mitigation result using decision tree it can analyze that particular fault can be predicted by its FTA (Fault Tree Analysis). For particular error there is particular mitigation technique defined by ETA (Event Tree Analysis) for one of them. By this we can predicate the fault by FTA and mitigate the fault by mitigation techniques. Various mitigation techniques can be described in the given table.

From table 3, It has been mentioned the mitigation techniques for a particular error code. Such as example of error code 400 it will occur due to missing request parameter and by check-pointing mitigation technique, it will be handled. Which provides efficient resource usage and it is used for long time running applications and it will be detected an application fault. By this means it can be applied for every hardware error code (400, 401, 403, 405, 406, 409, 415) and software error code (500 and 502).

The particular error code can predict the error and by particular mitigation technique, mitigate the fault and recover the system. This provide system robustness and work of the system efficiently even the fault is occur in organization

TABLE III: Mitigation techniques for error code

Error Code	Error Name	Description	Reactive Mitigation technique	Description	Type of fault detected
400	Bad Request	missing request parameter	Check pointing	Effective for long running applications, Provides efficient resource utilization	Application failure
401	Unauthorized	An invalid element token	Block	Block the request, Notify the user for valid request, More resource utilization	Process failure
403	Forbidden	Access to the resource is forbidden	Retry	Job is retried, Time inefficient	Host, Network failure
405	Method Not Allowed	Incorrect HTTP verb used	Rescue Workflow	Workflow is continued until task is failed.	Node, Application failure
406	Not Acceptable	The response content type does not match	Check pointing	Effective for long running applications, Provides efficient resource utilization	Application failure
409	Conflict	A resource is already exists	S-Guard	It is less stream processing, rollback recovery	Application, Node failure
415	Unsupported Media Type	The server cannot handle Content-Type	Job Migration or Load balancing	Job can be migrated to different machine.	Application, Node failure
500	Server Error	Something went wrong on the Cloud Elements server	Job Migration	More resource utilization, Time efficient	Application, Node failure
502	Failed Request	Endpoint provider was not able to perform a request	Task Resubmission	Job is retried on same or different resubmission resource	Application, Node failure

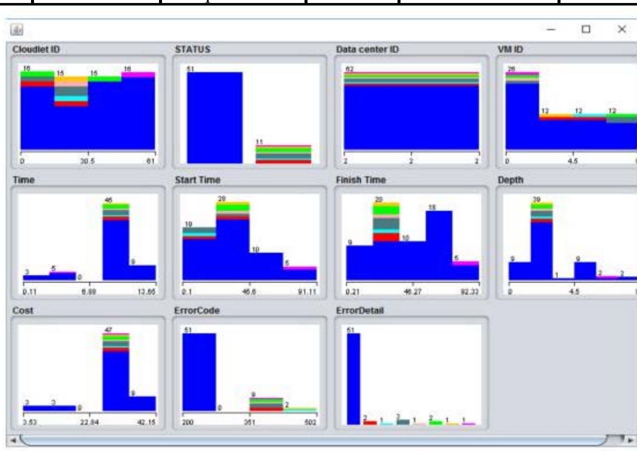

Fig. 8: Graph: error vs. various parameter

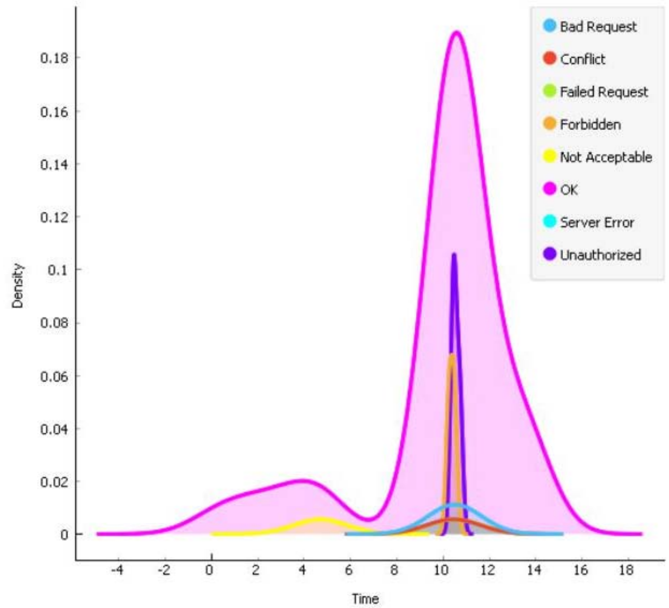

Fig. 9: Graph: execution time vs. Density

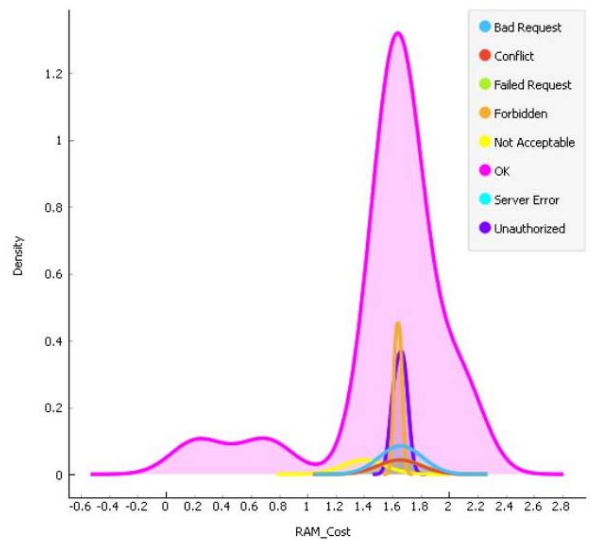

Fig. 10: Graph: RAM cost vs. Density

From the graph we can conclude the results. That shown the comparison of various error parameters. By figure 7, graph Showing the error rate which depends on several parameters such as cloudy, VM id, STATUS, error code, execution time, Depth. In figure 8, graph described density VS execution time. Which is concluded that the maximum error rate in particular execution time. In figure 9, graph described density VS RAM cost. Which is concluded that the maximum error rate in particular RAM cost. So, by this results we can concluded that maximum time, maximum usage of RAM and its cost and when the error occurs into the system. We can also predicted which error will be occur next and mitigate that fault using mitigation techniques.

VIII. CONCLUSION

Fault prediction and mitigation used to provide system availability and robustness when system have hardware or software fault. This literature review focused on the various fault prediction and mitigation techniques and tools used for implementing it and compare this technique and provide the best solution for fault prediction and mitigation.

IX. FUTURE WORK

Identifying the other failures apart from hardware and software as omission, network, response, crash, and mitigating their effects. Find techniques for recover that fault and make the system robust.

X. REFERENCES

[1] P. K. Patra, H. Singh, and G. Singh, "Fault tolerance techniques and comparative implementation in cloud computing," International Journal of Computer Applications, vol. 64, no. 14, 2013.

[2] H. Agarwal and A. Sharma, "A comprehensive survey of fault tolerance techniques in cloud computing," pp. 408–413, 2015.

[3] P. D. Kaur and K. Priya, "Fault tolerance techniques and architectures in cloud computing-a comparative analysis," pp. 1090–1095, 2015.

[4] K. Chauhan and V. Prasad, "Distributed denial of service (ddos) attack techniques and prevention on cloud environment," International Journal of Innovations & Advancement in Computer Science, pp. 210–215, 2015.

[5] A. Tchana, L. Broto, and D. Hagimont, "Approaches to cloud computing fault tolerance," pp. 1–6, 2012.

[6] R. Jhawar, V. Piuri, and M. Santambrogio, "Fault tolerance management in cloud computing: A system-level perspective," IEEE Systems Journal, vol. 7, no. 2, pp. 288–297, 2013.

[7] A. Ganesh, M. Sandhya, and S. Shankar, "A study on fault tolerance methods in cloud computing," pp. 844–849, 2014.

[8] M. N. Cheraghlou, A. Khadem-Zadeh, and M. Haghparast, "A survey of fault tolerance architecture in cloud computing," Journal of Network and Computer Applications, vol. 61, pp. 81–92, 2016.

[9] P. R. S. D. o. C. S. E. A. d. T. U. G. I. Dilip Kr Baruah, Lakshmi P. Saikia, "A review on fault tolerance techniques and algorithms in cloud computing environment," International Journal of Advanced Research in Computer Science and Software Engineering, May-2015.

[10] A. Bala and I. Chana, "Fault tolerance-challenges, techniques and implementation in cloud computing," IJCSI International Journal of Computer Science Issues, vol. 9, no. 1, pp. 1694–0814, 2012.

[11] V. K. Prasad, "Load balancing and scheduling of tasks in parallel processing environment,"

[12] D. Mittal and N. Agarwal, "A review paper on fault tolerance in cloud computing," pp. 31–34, 2015.

[13] A. K. HARPREET KAUR, "A survey on fault tolerance techniques in cloud computing," International Journal of Science, Engineering and Technology, 2015.

[14] N. Singla and S. Bawa, "Priority scheduling algorithm with fault tolerance in cloud computing," International Journal, vol. 3, no. 12, 2013.