INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 5)
Available online at: www.ijariit.com

Machine learning based approach for fault tolerance in cloud computing

Abhishek Gupta
abhishek.gupta0514@gmail.com
Bharati Vidyapeeth's College of Engineering, New Delhi, Delhi

Aashish Mamgain
aashish.mamgain79@gmail.com
HMR Institute of Technology and Management, New Delhi, Delhi

ABSTRACT

Cloud computing is a rising area that is currently engaged towards many of the IT industries. Bearing with the cloud architecture is a difficult challenge for us. This challenge can be achievable by fault tolerance and monitoring. Methods /Statistical Analysis: Fault Tolerance (FT) facilitates the process or the component to work smoothly even though in the occurrence of the failure. Monitoring is a procedure in which the fault is predicted before it occurs. Proactive and reactive measures can take place to run the cloud environment with tolerance in the failure occurrence. Reviewing the potential of FT and monitoring services is to make out the technique that serves in certain purpose. Fault tolerance systems are important for both providers of cloud services and customers. Findings: Based on this idea, this paper provides the different or diverse fault tolerance and monitoring mechanism to improve the reliability in a cloud environment. Applications/Improvements: It presents the information about the various techniques and methods used in the FT and also a future research direction in cloud FT.

Keywords— *Fault tolerance, Faulty node, Cloud computing, Proactive technique, Machine learning, Naïve bayes*

1. INTRODUCTION

Cloud Computing is an online service that helps in maintaining, manipulating, managing, and backing up of data remotely. Services are provided to the uses, by the means of various layers of abstraction and deploying different models, across the internet in cloud computing. The cloud computing concept can be understood as the distribution of various applications as services to the people and it can range from personal (small) domain such as users hosting their work to large public domains such as any enterprise which outsources its IT infrastructure to some external data centers.

Fig. 1: Fault occurrence

An example of the cloud computing services is Google's Gmail; many major corporations such as Google, Intel, IBM, Amazon, Oracle etc. offers a variety of cloud-based services that are used to find the solutions for day to day problems. Although cloud computing and its services are increasingly being adopted in the industry today still there are many serious subjects in it that demand our attention and research on the immediate basis. As for the figure 1 Fault tolerance is taken as one of the most critical and crucial subjects that need to be handled immediately and carefully as soon as possible.

The cloud system failure occurs because of the error caused by the fault which occurred in the system. Fault tolerance prevents the software failure that is occurred during the time of process execution. Whenever there is a chance of occurrence of the fault in the system, the fault management system handles with the system to prevent the failure of the system. Since the implementation of the fault tolerance is on the cloud, due to this advantage there will be cost reduction and increase the performance from failure. This paper intends to provide a better conceptual understanding of cloud computing and discusses the techniques and algorithms used in the fault tolerance domain. In this paper, we proposed a model to detect fault tolerance in cloud computing using machine learning based approach.

1.1 Background

Numerous faults occur in cloud computing environment every day. These faults can be resolved through the use of fault tolerance techniques, under fault tolerance policies which are discussed further in the paper.

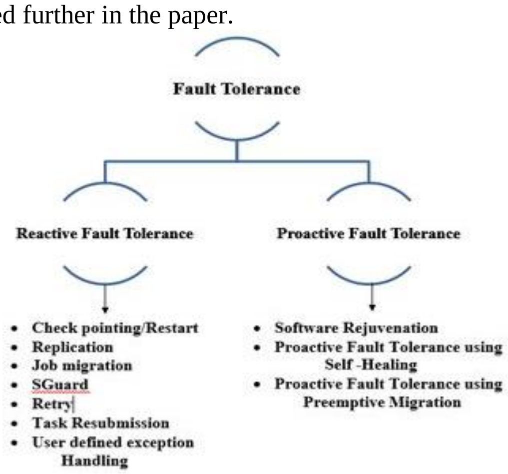

Fig. 2: Techniques for fault tolerance

1.2 Strategies under reactive fault tolerance

The strategies used for tolerance of fault helps in regulating the cause and effect of failures on the application, these techniques very helpfull in our cloud computing system on account of failure occurrences. Based on these policies there are various techniques listed as under:

- **Check pointing/ Restarting:** In this fault tolerance technique, when a failure in a task occurs due to some fault, this technique is provisioned to restart from a recently checked state. This type technique is more efficient for application those run for a long time.
- **Resubmission of Task:** In this, when a failed task occurs, the re-submit of that task is done to the same previous resource or to some dissimilar resource. This strategy has its main application in the domain of fault tolerance [1].
- **Job Migration:** In this approach, when a task fails, that failed task is carried out to another machine and this process is known as job migration. This strategy can be implemented using the HAProxy software.
- **User-defined exception handling:** In this, the users themselves take measures for the treatment of a failed task due to the occurred fault. [1].
- **Replication:** What happens in this strategy is that many replicas belonging to a particular task run concurrently on various platforms for successful completion of the task until all the replicated tasks crash.
- **Retry:** In Retry technique it retries the tasks that cause the fault on the cloud platform to eliminate the cause of failure when a fault occurs in the cloud environment.
- **Rescue workflow:** This technique allows the process to continue its execution even after the failure has occurred and this policy continues up to the point where for the process it is no longer possible to proceed further without mending or recovering the task that failed in between [1].

1.3 Proactive fault tolerance

This involves replacing the failure component by predicting the erroneous and faults etc. That is most prone to failure with the alternative components that are in working condition. Some of the method based on this concept are followed:

- **Rejuvenation of software:** In this technique, a system is designed in a way that system is restarted with the clean state after the periodic reboot this is known as fault tolerance technique. The main disadvantage of this concept is that if it is practiced too many times then the cost reserve unnecessarily. It also fails, if the system is designed to practice at long intervals then there’s a notable risk of the system fail in the cloud [1].
- **Self-Healing:** This technique deals with a large number of instances that run on virtual machines by automatically handling the failure of system instances.
- **Preemptive Migration:** In this, we preemptively take away the currently executing application from the nodes which are about to fail using preemptive migration. The process execution is constantly monitored throughout the cycle. While the application is being migrated, first its state is saved and then migration to a different node takes place.

2. RELATED WORK

Fault tolerance techniques for cloud computing id developing rapidly. This section consists of recent ideas and concepts in the field of fault tolerance as discussed in various papers published in various journals.

Table 1: Related work

Method	Description	Reference
Proactive fault tolerance using preemptive migration	Preemptively drift away from the currently executing application from the nodes which are about to fail	C. Engelmann, G. R. Vallee, T. Naughton, and S. L. Scot[2]
Fault Tolerance techniques and algorithm in cloud computing	Reviewed various fault tolerance techniques that can be implemented in the real-time distributed system	Arvind Kumar et al. [3]
Adaptive fault tolerance in real-time cloud computing environment	The adaptive behavior of the reliability weights that were assigned to each processing node and adding and removing them on the basis of their relatable reliabilities	Sheheryar Malik and Fabrice Huet[4]
BFTCLoud: A byzantine fault tolerance framework for voluntary-resource cloud computing	For cloud which provides high consistency in the cloud which also ensures high performance of these systems.	Yilei Zhang, Zibin Zheng and Michael R. Lyu [5]
Analysis of fault tolerance approaches in dynamic cloud computing	Proposed a load balancing algorithm in fault management domain of cloud computing	N. Chandrakala and P. Sivaprakasa[6]
Fault Tolerance techniques and comparative implementation in cloud computing	Provided catalog of different fault tolerance techniques based on various performance metrics	P. K. Patra, H Singh, G Singh[7]
Comprehensive survey of fault tolerance techniques in cloud computing	Discusses various fault tolerance techniques which are deployed according to their policies and applications	Himanshu Agarwal and Anju Sharma[8]
Fault tolerance in cloud computing: A review	Presents a review on fault tolerance in cloud comp. and discusses the reliability assessment algorithm and its impact analysis	Gagan Amrit Kaur and Sonia Sharma,[9]

3. PROPOSED MODEL

The proposed model falls in the bracket of Proactive Fault Management techniques. We can make the cloud computing system to undergo several rounds of testing and also observe and note the nodes which fail with every next testing cycle. Prediction algorithms or other specific classification algorithms can be applied in order to detect the fault in the node as other factors such as load and infrastructure that may affect failure characteristics of the node to predict and identify the nodes which are most probable of failure based on collected information.

After identifying the probable faulty node we could eliminate them replace them or even create a backup node which replaces the faulty node in an occurrence of the failure so that the cloud system keeps on working without any interruption. This model improves the reliability of the cloud computing system and hence does the job of fault management quite perfectly.

3.1 Classification algorithm

The Naïve Bayes classifier is the simplest machine learning algorithms which are used to classify the node in the cloud. It is based on the Bayes' theorem in mathematics which assumes strong independence between the attributes or features (predictors). A Naïve Bayesian classification model is particularly handy for very large datasets as it requires comparatively little efforts for its build and it does not have a complicated calculation or repetitive parameter figuring. Naïve Bayes' classifier, it is also deployed algorithm because it often outcastes the other complex and revised classification algorithms and does its job quite well.

This assumption used is called the class conditional independence. The Bayes theorem is used to calculate the posterior probability P(a|b) using from P(a), P(b) and P(b|a), which is the probability of failure of the node.

$$P(a \mid b) = \frac{P(b \mid a) \times P(a)}{P(b)} \quad (1)$$

Where,
P(a|b) = Posterior probability
P(b|a) = Likelihood
P(a) = Class Prior Probability
P(b) = Predictor prior Probability

For our cloud computing environment, we have:
P(a) = probability of failure of nodes in the model.
P(b) = probability of system failure.
P(b|a) = probability of system failure, given node fails.

Once we are done with this, we can identify the nodes which are most probable of experiencing fault during a process execution. Therefore, beforehand we actually know which node may fail, we then apply a Proactive fault management technique to rectify the fault. This whole process will increase the reliability of our fault tolerance management system.

4. IMPLEMENTATION

To create a real-world-like the scenario we did a simulation of cloud computing environment by creating several web pages (acting as server or nodes) and then identifying faulty nodes among them and we did this identification using MATLAB simulation. We took 15 web pages (nodes) and deployed the following stated algorithms to identify faulty nodes and do the proceedings thereafter.

4.1 Algorithm 1

According to the algorithm, each server consists of two parts of the code, if the first part is not working then the code of the second part is invoked. The second part generates the prime number. So, with the help of our algorithm, we can detect servers returning prime number and classify them as faulty. We consider all combinations possible between our 15 nodes and if a software fault occurs it is visualized using the algorithm deployed in MATLAB.

Step 1: Enter the number of servers;
Step 2: Make matrix of servers using meshgrid command;
Step 3: For random server(i), calculate i prime or not;
Step 4: Plot using surfc
Server (i, j)=simulation (i, j);
Step 5: If Server(i,j) faulty then i and j prime numbers;
Step 6: Else if Server(i,j)not faulty
if atleast one is composite;

Here, SERVER (i, j) = simulation (i, j) where 1<i, j<15, thus each cell in simulation shows two servers i and j connected in fashion i to j. Detection of the faulty server - after reading the 15 x 15 stimulation and selecting any cell, if it has a peak then it is a faulty server. For example - let's take cell (5,5), as we can see from the simulation graph that it's a peak so it is a faulty server.

Detection of the non-faulty server - after reading the 15x15 simulation graph and selecting any cell, if it does not show a peak then it is a non-faulty server. For example, let's take simulation (4,5), as we can see from the stimulation that it does not have a peak so it is a not a faulty server.

4.2 Algorithm 2

This algorithm works on the concept of hamming code. It works in the following manner:

Step 1: Server assigned 8 bits;
Final number of bits 8bits + 4check bits;
Step 2: c1=rem (a(1) + a(2) + a(4) + a(5) + a(7), 2);
Similarly calculate c2, c4 and c8;
Step 3: if server faulty, any of the bit will change;
Else bits remain same
Step 4: calculate new_checkbits
Step 5: perform xor operation (old_checkbits xor new_checkbits);
Step 5: check result of xor operation
Step 6: if 1 occurs at some place,
then server faulty
Step 7: convert xor_result decimal to get error bit position
Step 8: else if 1 not detected,
then server no-fault;

With the MATLAB simulation, we see the black line on the 7th-bit position signifying that there occurred a change in it so this was identified and the respective node was classified as being faulty.

So, we can do the identification of the nodes which are most probable of experiencing the fault and based on other conditions and on this collected dataset we can apply Naïve Bayes algorithm to predict beforehand which nodes are prone to experience faults and then apply Proactive strategies to make a strong Fault-tolerant system. We can measure reliability with MTBF factor or mean time between failures which is defined as the average time interval elapsed before a component fails and demands services. MTBF can be calculated using the equation:

$$\text{MTBF} = \frac{\text{Total time elapsed - Sum of downtime}}{\text{Number of failures}}$$

Table 2: Nodes and their fault status in the simulation

Index	Node	Fault
1	1	1
2	2	1
3	3	1
4	4	0
5	5	1
6	6	0
7	7	1

Column NODES represents node 1-15 and Column FAULT has 0 for no fault and 1 for fault occurrences in nodes.

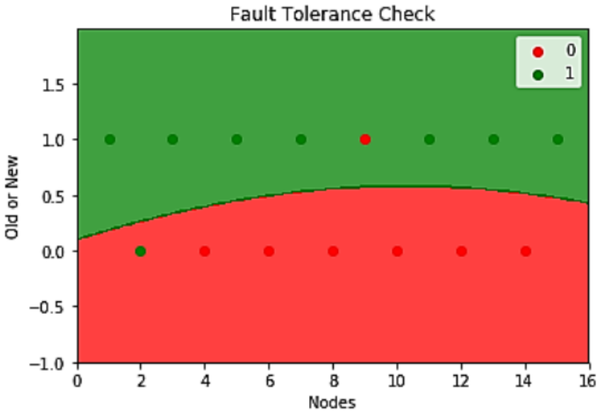

Fig. 3: Naïve Bayes implementation plot on the dataset

Figure 3 is a Naïve Bayes implementation plot on the dataset. The green dots are the nodes having a fault and the red dots are the nodes which do not experience any fault. The Naïve Bayes classifier divides the graph into two regions, green covers nodes that are fault-prone and red covers the working nodes. The accuracy of this prediction is almost 87%.

5. CONCLUSION

The cloud computing market is only going to grow as per current trends and in such a case when these dynamic cloud computing environments are more or less prone to failure, effective fault tolerance techniques are a must to ensure reliability over these cloud systems as their failure could be catastrophic. With this proposed model we worked in Proactive fault management domain whereby we first predict the nodes which are most prone to experiencing the fault with the
Naïve Bayes classifier and after that apply the fault tolerance techniques to ensure the enhanced reliability of the system. With our proposed model, we can enhance the reliability of the system, reliability values are used to estimate the failure probability node. Then the MTBF factor is used to measure the reliability of the system. With the application of Naïve Bayes, we can reduce the numbers of node failures with the accuracy of nearly 87%. There is a scope of using other classification algorithms for the same and then it can be compared. Further, scalability, which is a QoS attribute, can be tested to enhance the system and reduce the overhead.

6. REFERENCES

[1] Anju Bala and Inderveer Chana, "Fault Tolerance Challenges, Techniques and Implementation in Cloud Computing", January 2012.

[2] C. Engelmann, G. R. Vallee, T. Naughton, and S. L. Scott, "Proactive Fault Tolerance Using Preemptive Migration", May 2009.

[3] Arvind Kumar et al., "Fault Tolerance Techniques and Algorithms in Cloud Computing", 2011.

[4] Sheheryar Malik and Fabrice Huet, "Adaptive Fault Tolerance in Real-time cloud computing", IEEE, July 2011.

[5] Yilei Zhang, Zibin Zheng, and Michael R. Lyu, "BFTCLoud: A Byzantine Fault Tolerance Framework for Voluntary-Resource Cloud Computing", September 2011.

[6] N. Chandrakala and P. Sivaprakasam, "Analysis of Fault Tolerance Approaches in Dynamic Cloud Computing", February 2013.

[7] P. K. Patra, H Singh, G Singh, "Fault Tolerance techniques and comparative implementation in cloud computing", International Journal of Computer Applications, 2013.

[8] Himanshu Agarwal and Anju Sharma, "A Comprehensive Survey of Fault Tolerance Techniques in Cloud Computing", 2016.

[9] Gagan Amrit Kaur and Sonia Sharma, "Fault Tolerance in Cloud Computing: A Review", IJCST, April-June, 2017.

[10] Sourabh Dave and Abhishek Raghuvanshi, "Fault Tolerance Techniques in Distributed System", International Journal of Engineering Innovation & Research Vol. 1, Issue 2, ISSN: 2277 – 5668, 2012.

[11] Prasenjit Kumar Patra, Harshpreet Singh &Gurpreet Singh, "Fault Tolerance Techniques and Comparative Implementation in Cloud Computing" International Journal of Computer Applications, Vol. 64, No.14, February 2013.

[12] Prasenjit Kumar Patra, Harshpreet Singh &Gurpreet Singh, "Fault Tolerance Techniques and Comparative Implementation in Cloud Computing" International Journal of Computer Applications, Vol. 64, No.14, February 2013.

