International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019

Reliability of Fault Tolerance in Cloud using Machine Learning Algorithm

S. Harini Krishna, G. Niveditha, K. Gnana Mayuri

Abstract—The basic fault tolerance issues seen in cloud computing are identification and recovery. To fight with these issues, so many fault tolerance methods have been designed to decrease the faults. However, due to the reliability and web based service giving behavior, fault tolerance in cloud computing will be a huge challenge. The present model is not just on tolerating faults but also to decrease the possibility of future faults as well[4].The fault tolerance deals with the exact and constant operation of the fault segments. The processing on computing nodes can be done remotely in the real time cloud applications, so there could be more possibilities of errors. Hence there lies an immense necessity for fault tolerance to attain consistency to the real time computing on cloud infrastructure. The "fault tolerance" can be explained through fault processing that have two basic stages. The stages are

(i) The effective error processing stage which is used to intended for carrying the "effective error" back to inactive state, i.e., before the error occurred

(ii) The latent error processing stage intended for guaranteeing that the fault does not get effective once again.

I. INTRODUCTION

Cloud computing provides numerous resources in the type of services to the end nodes on-need basis. It enables various businesses and users to utilize applications without installing them on their phones or laptops and allows the access to necessary resources over the Internet. It provides various features such as connectivity, high performance, reliability, pay-as-you-go, interactivity, ease of programmability, efficiency, scalability, elasticity and management of large amount of data and there by transforming Information Technology from a product to a service. [1]

Fault tolerance refers to a technique of system design that lets a system to keep working when one of its parts fails or it can be explained as a capacity of a system to react quickly to an unexpected equipment or programming break down. If the system is not completely operational, fault tolerance solutions may allow a system to continue operating at lower capacity rather than shutting down completely following a failure. [2]

Revised Manuscript Received on October 30, 2019.
***** Correspondence Author
S. Harini Krishna*, Assistant Professor, Department of Computer Science Engineering, Geethanjali College of Engineering and Technology, Hyderabad, India. (E-mail: harinishiva9@gmail.com)
G. Niveditha, Assistant Professor, Department of Computer Science Engineering, Geethanjali College of Engineering and Technology, Hyderabad, India. (E-mail: nive.gopigari@gmail.com)
K. Gnana Mayuri, Assistant Professor, Department of Computer Science Engineering, Geethanjali College of Engineering and Technology, Hyderabad, India. (E-mail: mayuri.787@gmail.com)

II. BENEFITS OF CLOUD COMPUTING

Cloud computing decreases the response time and running time of a task. It also lowers the risk in deploying application, lowering cost of deployment, and reducing the effort and increasing the innovation.

Increased Throughput: Cloud makes use of thousands of servers to complete anassignment in less time unit compared to the time taken by a individual server.

Decreases infrastructure risk: Cloud can be utilized by the organizations to minimize the load of buying physical servers. The issues of higher investment and deployment of servers depending upon the workload can be resolved by considering investment on infrastructure for the applications whose attainment is short-lived.

Minimize cost of entry: Various characteristics which are mentioned earlier reduces the cost for organizations to enter new markets:

The capital investment is minimized to zero by renting the infrastructure instead of buying it and there by controlling the cost.

The rapid application development helps to minimize the time taken to get into the market, thereby giving organizations an edge against the competition.

Focus on innovation: Organizations dealing with the issue to infrastructure deployment can now focus on innovating things. [2]

III. STRATEGIES UNDER REACTIVE FAULT TOLERANCE

The strategies used for fault tolerance helps in regulating the cause and effect of failures on the application, these methods were very help full in cloud computing system on instances of failure occurrences. Based on these policies there are various techniques[3] :

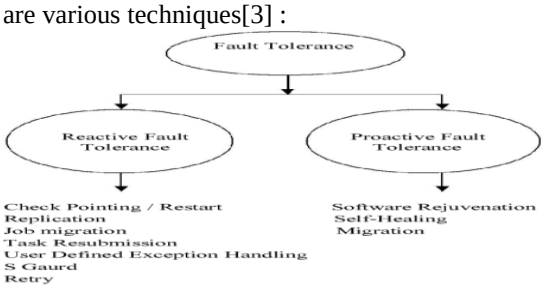

Fig 1: Fault tolerance techniques

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.

- *Task resubmission:*

According to this, when a task fails, the resubmission of that task is done to the same previous resource or to some other resource. This strategy has its major application in the domain of fault tolerance [3].

- *Job Migration:*

In this methodology, when a task fails, that failed task is carried out to another machine and this process is known as job migration. This task can be implemented using the HAProxy software.

User-defined exception handling:

In this process, the users themselves take measures for the treatment of a failed task which was occurred due to a fault. [3].

- *Replication:*

In this process, many replicas belonging to an individual task run simultaneously on various platforms for successful completion of the task until all the replicated tasks gets crashed.

- *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].

3.1. Proactive fault tolerance

It involves replacing the failed components by predicting the errors and faults etc, which are most prone to failures with the alternative components which are in working condition. Some of the methods based on this concept are as follows:

- *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 in this concept is that if it is run too many times then it costs huge and if the system is designed to practice at long intervals then there is a remarkable risk of the system failure in the cloud [3].

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 process, is preemptively taken away from the currently executing application from the nodes which are going to fail using preemptive migration. The process execution is thoroughly monitored throughout the cycle. While the application is being migrated, first its state is saved and then migration to a different node takes place.

IV. 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 easy to use for very large datasets as it requires comparatively less efforts for its buildup and it also don't have a complicated calculation or repetitive parameter figuring. Naïve Bayes' classifier is also a deployed algorithm because it often outcastes the other complex and revised classification algorithms and does its job very well[5].

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.

$p(a|b)=p(b|a)*p(a)/p(b)$.....(1)

V. IMPLEMENTATION & RESULTS

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 simulation. Here 10 web pages (nodes) were taken and following stated algorithms were taken to identify faulty nodes.

5.1. 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[6]. We consider all combinations possible between our 10 nodes

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

5.2. Algorithm 2

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

Step 1: Server assigned bits;
Final number of bits 7bits + 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_checksum
Step 5: perform modular 2 operation (old_checksum modular 2 new_checksum);

Step 5: check result of modular 2 operation
Step 6: if 1 occurs at some place,
then server faulty
Step 7: convert modular 2_result decimal to get error bit position
Step 8: else if 1 not detected,
then node no-fault;

VI. CONCLUSION

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 performance factor is used to measure the reliability of the system. By applying Naïve Bayes, the number of node failures can be reduced with the accuracy of nearly 90%. There is a scope of using other classification algorithms for the same and then it can be compared.

REFERENCES

1. Kumari, P., & Kaur, P. (2018). A survey of fault tolerance in cloud computing. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2018.09.021
2. 2. Amin, Z., Singh, H., & Sethi, N. (2015). Review on Fault Tolerance Techniques in Cloud Computing. International Journal of Computer Applications, 116(18), 11–17. https://doi.org/10.5120/20435-2768
3. 3. Kochhar, D., Kumar, A., & Hilda, J. (2017). an Approach for Fault Tolerance in Cloud Computing UsingMachineLearningTechnique. 17(22), 45–351. https://doi.org/10.13140/RG.2.2.31419.67366
4. 4 . Mayuri, K. G., Krishna, S. H., & Niveditha, G. (2018). Developing Fault Tolerance in Cloud Computing based on Machine Learning Approaches. 10(10), 7–11.
5. 5. Bala, A., & Chana, I. (2012). Fault Tolerance-Challenges, Techniques and Implementation in Cloud Computing. International Journal of Computer Science Issues, 9(1), 288–293.
6. 6. Gupta, A., Delhi, N., Mamgain, A., & Delhi, N. (2018). Machine learning based approach for fault tolerance in cloud computing. 4(5), 59–62