International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 22 (2018) pp. 15762-15765

A Classification Approach for Proactive Fault Tolerance in Cloud Data Centers

Ghamdan Mohammed Qasem, Dr. Madhu B. K.

Research scholar in Computer Science Department, Jain University,Bangalore, Karnataka, India.

Professor & Head, Dept of ISE, RRIT, Bangalore, Karnataka, India.

Abstract

Reliability and availability are the two greatest challenges faced by cloud data centers as it affects the service level agreement between the provider and consumer. Most of the existing approaches use checkpoint and reactive mechanisms which are not considered to be efficient due to performance issue and challenges. So an improved fault tolerance technique is more under research to solve the issue of fault tolerance. This paper proposes a Fuzzy min max neural network Classification approach that can predict failure in advance and solve the problem in case of overlapping. Bu initially training the data set and the threshold value is fed into the testing data set using the proposed Fuzzy min max neural approach. Results show that there is an improvement in the performance when compared over the previous approaches.

INTRODUCTION

Cloud computing is playing vital role in all fields of education, research areas, IT world. Cloud servers have failure of many virtual machines is a challenge that challenge will be solve by cloud data centers. The resources of the systems like memory utilization, CPU temperature, Bandwidth, and other computing resources may fail in particular amount of time. This failure of the resources may effect on system performance or system efficiency. To avoid this undesirable consequence, proactive failure tolerance approach is used. The main goal of the proactive failure tolerance approach is to find out the overlapped data , minimizes the time require for finding the failure of the virtual machines and finding the usage of the resources in particular system. The implementation of the proposed proactive failure tolerance approach is taking into consideration the threshold values from the training data set that is later applied to the testing data set. By using a new Bi-Level classifier based on Fuzzy Min-Max Neural Network (FMNN) a classifier is used for finding out whether a virtual machine may fail or not that depends on the values related with usage of the resources .KNN can be used for both classification and regression predictive problems. The proposed minimum distance based KNN classifier starts with the overlapped data as input which are resulted from the FMNN classifier. To classify the overlapped data, as failure or success the KNN classification algorithm is utilized.

LITERATURE SURVEY

Wang et al. proposed a fault tolerance that is considered as one of the important means that are used to achieve dependability and reliability in cloud computing [1].Sousa et al. proposed the essential preferences of completing adaptation to internal failure in cloud computing fuse bring down cost, and enhanced execution measurements [2]. Amoon and Mohammed mentions that fault tolerance consists of fault detection, backup, and failure recovery, and nearly all fault tolerance approaches are based on the use of redundancy [3].Sharkh et al. proposed an approach that faults can be handled on the basis of two types of fault tolerance policies. One is the Reactive fault tolerance that reduces the impact of failures when the failures have occurred. Other is Proactive fault tolerance that predicts faults before they occurred [4].

Bruneo et al. used a reactive fault tolerance techniques which often do not scale due to massive I/O requirements and relies on manual job resubmission [5].Jayasinghe et al. discuss about the scalability and complexity of reactive fault tolerance, research on system dependability which has recently shifted onto failure prediction i.e. proactive fault tolerance technologies [6].

Cheraghlo et al. justify that proactive approaches potentially reduce the failure rates within the cloud and also increase the capacity and throughput. An effective proactive approach should provide a way with all available information considered, to et al avoid any job from any possible failures [7].

Egwutuoha et al.proposed Software Rejuvenation, Preemptive migration, Self-healing that are some of the proactive fault tolerance policies involved in the literature that may predict faults and errors and get rid of the paralyzed components [8]. Zhu et al. proposed Software rejuvenation policy which involves terminating an application, cleaning up the system internal state and restarting it to prevent the occurrence of future failures [9]. Rimal et al. proposed a preemptive migration, continuous monitoring and analysis are done on the output according to the feedback from the output [10]. Buyya et al .proposed a self-healing policy and improvement is achieved without any replication and redundancy. With this policy, the failed versions of an application will be handled in multiple VMs [11].

Bahga et al .proposed in contrast to these existing policies have used machine learning algorithms such as Artificial

Neural Network, K-Nearest Neighbor for predicting the failures which provide more efficient resource consumption [12]

Bi-level Classifier*:* Fuzzy min max neural network Classifier

By using a new Bi-Level classifier based on Fuzzy Min-Max Neural Network (FMNN) , the classifier use the future values of resource utilization metrics to predict and classified them into two different classes namely failure and un failure. The proposed FMNN classification algorithm utilized in this research work is mentioned below as shown in algorithm 1.

Algorithm 1:

Input: X values of n number of VMs to classify, [$VM_i = 1 to\, n$]

Output: Failure/ Un-failure/ Unpredictable (Overlapping)

Step 1: set the values of U_T, L_T

Step 2: Check

IF $(X \geq U_T)$ // X is the average utilization of each VM

Step 3: create hyper box B_1 and label class K_1 as failure

Step 4: Check

If X belong to an existing hyper box B_1 with class label K_1

Put input data into B_1

end if

Step 5: Check

ELSE IF $(X \leq L_T)$

Step 6: create hyper box B_2 and label class K_2 as success

Step 7:Check

If X belong to an existing hyper box B_2 with class label K_2

Put input data into B_2

end if

Step 8:Else

Step 9: Label as unpredictable/overlapping data x

Step 10: End

This classifier results in an unpredictable case as shown in algorithm 3.1. This degrades the performance of the system. In order to overcome such drawback in this work, then introduce a minimum distance based KNN classifier in this research work.

- Minimum Distance based KNN Classifier

To classify the overlapped data as failure or success the KNN classification algorithm is utilized which is expressed below as given in algorithm 3.2 .

Algorithm 2

Input: Overlapped data x.

Output: Failure/Success

Step1: set the values of U_T, L_T

Step 2: Compute

$$d_U = U_T - x$$

$$d_L = x - U_L$$

Step 3: Check

IF $d_U < d_L$, Then label x as failure

Step 4: Else

Label x as Success

Step 5: End

RESULTS AND DISCUSSION

Cloud Sim with java is used to build and train the data set which is a free java based simulation tool that can use neural networks framework for building training data set. CloudSim supports lots of learning algorithms and has java APIs to build and train the network in cloud based applications. The network is trained with fuzzy neural network concept and then the threshold value is calculated based on the selected parameters. The combination of the parameters that can produce the closest value to the threshold is fed to the training data set.

The results discussed in Figure 1 to Figure 5 gives the Average utilization of each virtual machine as the CPU utilization, memory Utilization, Bandwidth Utilization and CPU temperature can be automatically calculated by CloudSim . Sample data is collected and processed for the training stage and the missing data values are removed and the data is adjusted in a correct format. In the training phase the network is given the input vector so that the output can find out the relationship between input and output values.

At the time of data collection, faults are injected into the system to create failure scenarios. Different workloads are fed and the fan and CPU speed and memory consumption is calculated and the time of virtual machine failure is recorded.

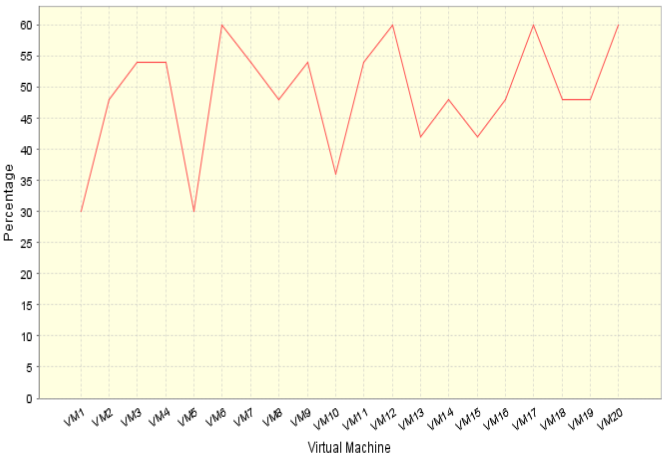

Figure 1: CPU utilization of 20 VMs

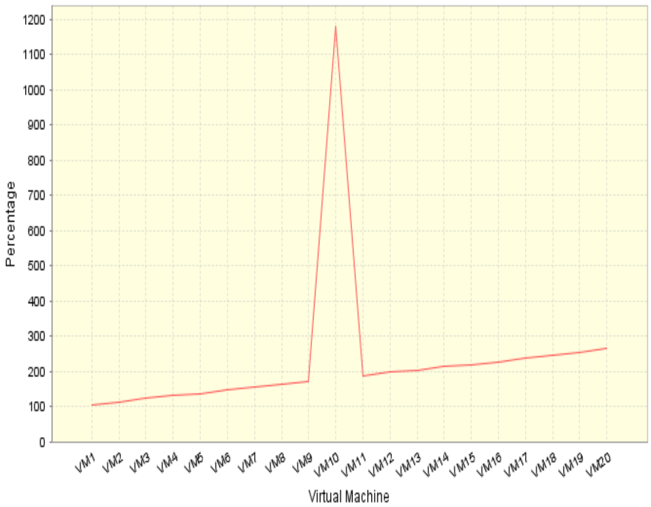

Figure 2: Memory utilization of 20 VMs

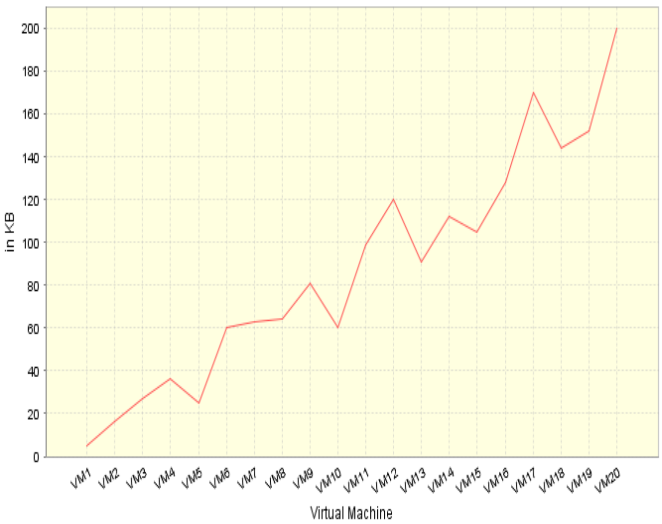

Figure 3: Bandwidth of 20 VMs

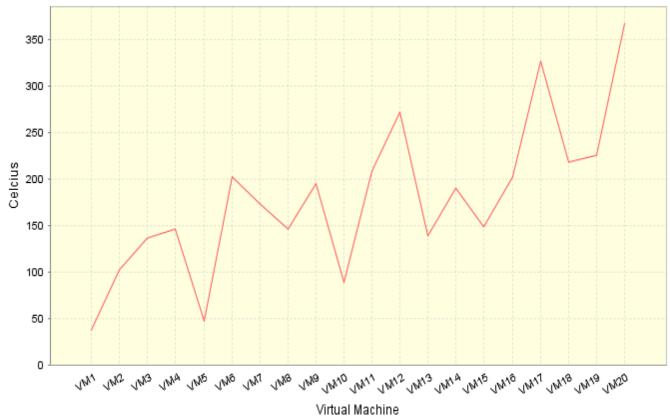

Figure 4: CPU temperature of 20 VMs

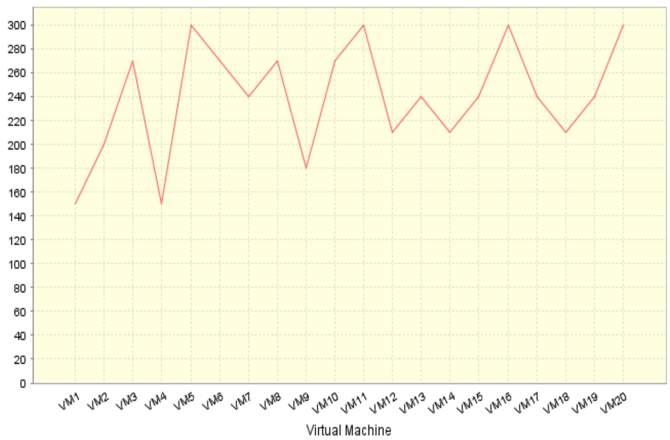

Figure 5: Failure time of 20 VMs

Calculating current simulation time and prediction time of each virtual machine

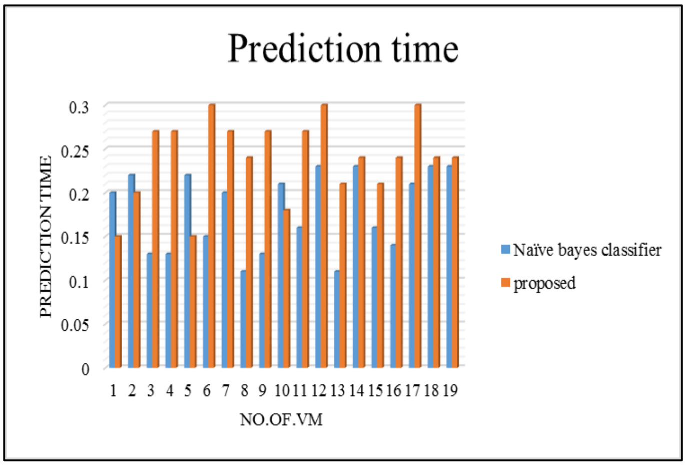

Figure 6: Prediction Time of VM Failure

The simplified graph in figure 6 discuss about the early prediction state of virtual machine failure. It is very clear that the parameters are responsible for the prediction of failure.

The proposed algorithm finds out the prediction rate and determines how early the failure can be predicted. This helps in early prediction and migrate the virtual machine to another server thereby helps in the achievement of reliability.

Table 1. Performance evaluation

	Proposed (FMM-KNN classifier)	Existing (Naive Bayes classifier)
PPV	0.00	0.00
Specificity	95%	80%
Accuracy	95%	80%
FDR	100%	100%

From the above table 1, it is seems to be that comparatively our proposed FMM-KNN classifier results high accuracy and prediction than the existing classifier such as Naïve Bayes classifier. So that can conclude that the proposed classifier will be a better choice of technique to employ in the cloud system which mainly has the need of early prediction of VM failure.

CONCLUSION

This proposed work uses the utilization of VM resources evaluation which depends upon the measurements, for example, CPU utilization, memory utilization, task scheduling capacity. By utilizing these measurements, a Bi-Level classifier Fuzzy Min-Max Neural Network (FMNN) classifier is used that predicts the future estimations of utilization of vm resources. It is then classified into two unique classes in particular, failure and unfailure. Thus the problem of overlapping is solved.

REFERENCES

[1] Wang, Chao, Frank Mueller, Christian Engelmann, and Stephen L. Scott. "Proactive process-level live migration and back migration in HPC environments." Journal of Parallel and Distributed Computing, Vol. 72, No. 2, pp. 54-267, 2012.

[2] Sousa, Paulo, Alysson Neves Bessani, Miguel Correia, Nuno Ferreira Neves, and Paulo Verissimo. "Highly available intrusion-tolerant services with proactive-reactive recovery." IEEE Transactions on Parallel and Distributed Systems, Vol. 21, No. 4, pp. 452-465, 2010.

[3] Amoon, Mohammed "A fault-tolerant scheduling system for computational grids" Computers & Electrical Engineering, Vol. 38, No. 2, pp. 399-412, 2012.

[4] Sharkh, Mohamed Abu, Manar Jammal, Abdallah Shami, and Abdelkader Ouda. "Resource allocation in a network-based Cloud computing environment: design challenges." IEEE Communications Magazine, Vol. 51, No. 1, pp. 46-52, 2013.

[5] Bruneo, Dario, Salvatore Distefano, Francesco Longo, Antonio Puliafito, and Marco Scarpa, "Workload-based software rejuvenation in cloud systems" IEEE Transactions on Computers, Vol. 62, No. 6, 1072-1085, 2013.

[6] Jayasinghe, Malith, Zahir Tari, Panlop Zeephongsekul, and Albert Y. Zomaya. "Task assignment in multiple server farms using preemptive migration and flow control." Journal of Parallel and Distributed Computing, Vol. 71, No. 12, pp. 1608-1621, 2011.

[7] Cheraghlou, Mehdi Nazari, Ahmad Khadem-Zadeh, and Majid Haghparast. "A survey of FT architecture in Cloud computing." Journal of Network and Computer Applications Vol. 61, pp. 81-92, 2016.

[8] Egwutuoha, I. P., Levy, D., Selic, B., & Chen, S. (2013). A survey of FT mechanisms and checkpoint/restart implementations for high performance computing systems. The Journal of Supercomputing, 65(3), 1302-1326.

[9] Zhu, Jun, Zhefu Jiang, Zhen Xiao, and Xiaoming Li. "Optimizing the performance of VM synchronization for FT." IEEE Transactions on Computers, Vol. 60, No. 12, pp. 1718-1729, 2011.

[10] Rimal, Bhaskar Prasad, Admela Jukan, Dimitrios Katsaros, and Yves Goeleven. "Architectural requirements for Cloud computing systems: an enterprise cloud approach." Journal of Grid Computing, Vol. 9, No. 1, pp. 3-26, 2011.

[11] Buyya, Rajkumar, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." Future Generation computer systems, Vol. 25, No. 6, pp. 599-616, 2009.

[12] Bahga, Arshdeep, and Vijay K. Madisetti. "Analyzing massive machine maintenance data in a computing cloud." IEEE Transactions on Parallel and Distributed Systems, Vol. 23, No. 10, pp. 831-1843, 2012.