Proactive Fault Tolerance in Cloud Data Centers to Improve Performance

Ghamdan Mohammed Qasem¹, Madhu B. K.²
¹Research scholar in Computer Science Department, Jain University, Bangalore, Karnataka, India
²Professor & Head, Dept of ISE, RRIT, Bangalore, Karnataka, India

Abstract: Cloud computing has become popular as it provides scalability and flexibility for high performance applications by provisioning of virtual machines. As the number of virtual machines grows large, any fault during virtual machine execution will lead to performance degradation of applications. Most of the existing works are not considering the anticipation of virtual machine failure and in advance and initiate necessary action as soon a failure is predicted. In this paper, A Fuzzy min max Neural Network classification approach is proposed to predict virtual machine failure by monitoring the deteriorating status of virtual machines. A minimum distance based KNN classifier is used in classifying the overlapped hyper box. The proposed method will improve the FMNN and thereby yielding lower computational complexity and faster process. The experimentation results obtained by simulation prove that virtual machine failures can be anticipated in advance. After a failure is predicted, a decision is made to do a migration from a deteriorating node to a spare node. Hence the proposed algorithm will generate more accurate prediction with high convergence speed.

Keywords: Proactive, Virtualization, Optimization, Migration

I. INTRODUCTION
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. 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.

II. PROBLEM STATEMENT
Most of the existing techniques for proactive fault tolerance make use of Fuzzy Min-Max Neural Network (FMNN) Classifier for predicting the VM Failure in advance. FMNN generates a common problem called overlapping. The existing techniques makes use of compensatory neurons in neural network but this will increase the complexity of the system. The improved FMNN is implemented by adding some rules for overlapping to detect possible overlapping areas. This will cause a greater number of hyper boxes and leads to additional pruning operation because when there is less number of hyper boxes, the process will be faster. Therefore, a new method is proposed to predict fault tolerance efficiently.

III. BRIEF SURVEY OF EARLIER WORK
The authors in [1] proposed a Fault Tolerance (FT) technique for handling faults proactively in HPC systems to condense the barricade chronometer implementation instance in the occurrence of responsibilities and enlarged a standard FT algorithm. The proposed algorithm did not depend on an auxiliary node earlier to calculate or disintegrate. The authors scrutinized the dollar price of provisioning standby nodes in order to review the importance of the technique and investigational outcome acquired from an original cloud finishing location illustrated that...
implementation of time by the wall clock and the calculation exhaustive relevancies in cloud can be concentrated via maximum upto 30%. The occurrence of check pointing of calculation demanding applications can be reduced to 50% only with fault tolerance technique for HPC systems, measured up to existing FT techniques. Zeeshan Amin et al. [2] an assortment of fault recognition schemes and architectural models had been anticipated to enlarge fault tolerance capability of cloud. The most important rationale of this paper was to recommend an algorithm via non-natural Neural Network for burden recognition which conquer the gaps of formerly executed algorithms and afford a fault tolerant model.

Dr. Chandralekha et al. [3] explored the key relationships between error tolerance and organization recital and developed metrics to measure fault tolerance within the context of system performance. By applying them to a sample mobile cloud computing environment consisting of multiple mobile nodes and demonstrate the usefulness of design metrics such as robustness. In addition, robustness was optimized using genetic algorithms and the effective fault tolerance presented by the system was evaluated. Taskeen Zaidi et al. [4] deals with the thoughtful of fault tolerance procedures in cloud environment and association with diverse models on different parameters had been completed. Fault tolerance procedure was studied with the assist of well recognized object-oriented language united Modeling Language and situation diagrams were designed and legalized via the perceptions of restricted State Machine.

Jhawar et al. [5] found an innovative system level modular perspective on the creation and management of cloud fault tolerance and proposed all inclusive sophisticated approach to sharing application developers and users with implementation details of fault tolerance technology using a dedicated service layer. In particular, users can themselves mention and apply the required level of fault tolerance at the service layer, and no knowledge of the assumed cloud and the fault tolerance technology available for that implementation was necessary. Kandaswamy et al. [6] introduced an efficient algorithm based on over provisioning replication as the main mechanism for fault tolerance while scheduling or mapping workflow tasks onto different resources. In over-provisioning, multiple copies of all workflow tasks (same input data-set) are executed parallelly. The aim is to maximize the success rate for the workflow task that is if one copy gets failed, other copies may be used. Nurmi et al. [7] and Schroeder et al. [8] introduced a resource reliability model for fault tolerance using Waybill distribution and taking „mean time between failures (MTBF)“ as a parameter on high performance clusters. In this model, it is reported that if the shape parameter comes less than 1, then it means that the hazard rates (the frequency with which a system or component fails) decrease with time. Reed et al. [9] identified all the reliability challenges related to large-scale HPC systems and provided with the adaptive techniques of failure detection (using performance contracts) and runtime adaptation for MPI applications. Lu et al. [10] used fault injection approach to study fault sensitivity in MPI applications.

Khalili et al. [11] studied the reliability of computational grids by gathering data or via deployment of monitoring infrastructure on two production grids, one of which is the Tera Grid. Inference that they have drawn was the success rates of application lies between 55% and 80%. Therefore they were motivated to include fault-tolerance mechanisms for workflows executing on computational grids. Hwang et al. [12] presented a model on failure detection service (based on notifications) and a scalable framework for tolerating grid failures using simulation as the evaluation parameter.

Alonso et al. [13] presented issues on fault-tolerance for commercial workflow management systems (WFMS) like Current commercial WFMS lack in fault tolerance features and techniques borrowed from transactional systems like replication should be used for increased availability and better exception handling. Chandra and Toug [14] proposed an improvement of this classic heartbeat implementation. In the proposed algorithm, the process q (monitoring process) uses a sequence of fixed time points T1, T2, T3...called freshness points in order to determine whether to suspect the process p. The freshness point Ti is an estimation of arrival time of heartbeat from p. The advantage of this algorithm is that detection time is independent from the last heartbeat message, thus increasing accuracy of the failure detector as it avoids premature timeout.

The authors in [15] proposed Low Latency Fault Tolerance (LLFT) Model that utilizes leader/follower replication approach and provides fault tolerance for distributed applications deployed within a cloud computing environment. The novel commitments of the LLFT middleware incorporate the low Latency Messaging Protocol, the leader-determined membership protocol and the virtual determinate Framework.

The work at [16] puts forward a dynamic adaptive fault tolerance strategy (DAFT) that is focused around the standards and semantics of cloud fault tolerance. An analysis on relationship between different failure rates and two different fault tolerance techniques, check-pointing and replication has been carried out. A dynamic adaptive model has been built by combining the two fault tolerance models which helps to increase the serviceability.
IV. METHODOLOGY

4.1 Proactive fault tolerance

The proposed method begins with the evaluation of VM resource utilization which depends on the metrics such as CPU utilization, memory utilization, job capacity and CPU temperature. By using these metrics, a new Bi-Level classifier based on Fuzzy Min-Max Neural Network (FMNN) classifier is proposed that predicts the future values of resource utilization metrics and classified them into two different classes namely failure and un failure. In that there is a difficulty in classifying the overlapped hyper box which is solved by further introducing a minimum distance based classifier. The minimum distance based KNN classifier classifies the overlapped hyper boxes into one of the above two classes by calculating the minimum distance between them. Our proposed Bi-Level FMNN-KNN Classifier is illustrated in figure 2. Each class from Bi-Level FMNN-KNN classifier is compared with a threshold value and depends upon the threshold they are categorized as Failure VM and Un failure VM. The failure VMs are migrated to newer location to save the resources. The newer location of each Failure VM is selected optimally using Modified Bat optimization. Figure 1 demonstrates the process flow of our proposed proactive fault tolerance.

![Figure 1: Process flow of proactive fault tolerance](image1)

![Figure 2: Bi-Level FMNN-KNN Classifier](image2)
The proposed system is implemented in java cloudsim platform and the results obtained are compared with existing researches in terms of failure prediction time.

4.2 Pseudocode for Fuzzy Min Max Classifier

IF \( \bar{X} \leq UT \) // \( X \) is the average utilization of each VM
create For VM i \( \bar{U} \) to n do
hyper box B1 and label class K1 as failure
□ If \( X \) belong to an existing hyper box B1 with class label K1
Put input data into B1
end if
ELSE IF \( \bar{X} \lt LT \)
create hyper box B2 and label class K2 as success
If \( X \) belong to an existing hyper box B2 with class label K2
Put input data into B2
end if
End ELSE IF
ELSE Check for overlapping data x

4.3 Pseudocode for KNN classifier

\( \bar{X} \) for x \( \bar{X} \) to m do //\( x \) is the overlapped data and m is the number of overlapped data

\( \bar{X} \) Compute distance d min \( \bar{U} \), LT, \( x \)
\( dU \bar{U} \) UT \( dL \) x
\( dL \bar{x} \) U L

IF \( dU \) \( \bar{U} \) Then label x as failure
ELSE IF \( dU \) \( LT \) Then label x as success
ELSE Label x as failure
End ELSE
End ELSE IF
End IF
End for

Prediction time calculation

Prediction time = tEF - Current Simulation Time

The current simulation can be calculated by the cloudsim

By using this we can calculate the prediction time of the proposed work.
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 2.

Algorithm 1:
Input: X values of n number of VMs to classify, \( [\text{VM}_i]^{1 \to n} \)
Output: Failure/ Un-failure/ Unpredictable (Overlapping)
Step 1: set the values of UT , LT
Step 2: Check
\( \text{IF} \quad X \quad \text{UT} \quad \text{IF} \quad X \quad \text{is the average utilization of each VM} \)
Step 3: create hyper box B1 and label class K1 as failure
Step 4: Check
If \( X \) belong to an existing hyper box B1 with class label K1
Put input data into B1
end if
Step 5: Check
ELSE IF \( \text{X} \quad \text{LT} \quad \text{ELSE IF X is the average utilization of each VM} \)
Step 6: create hyper box B2 and label class K 2 as success
Step 7: Check
If \( X \) belong to an existing hyper box B2 with class label K 2
Put input data into B2
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.

4.4 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.

Algorithm 2
Input: Overlapped data x.
Output: Failure/Success
Step 1: set the values of UT, LT
Step 2: Compute
\[ d_U(x) \]
\[ d_L(x) \]
Step 3: Check
IF \( d_U \neq d_L \), Then label x as failure
Step 4: Else
Label x as Success
Step 5: End

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.

V. RESULTS AND ANALYSIS

The proposed bi level FMNN-KNN classifier overcomes the drawback present in the previous research work such as classification of unpredictable/overlapping data which is resulted from the FMNN classifier. This will improve the performance of proposed method and thereby yielding lower computational complexity and faster process. Hence the proposed algorithm will generate more accurate prediction with high convergence speed as shown in

![Figure 3: Prediction Time of VM Failure](image)
fig3, fig4, fig5, fig6, fig 7. Even though there are numerous classifiers in the literatures here we are utilizing Fuzzy Min-Max Neural Network (FMNN) Classifier for predicting the VM Failure. The purpose of selecting FMNN classifier is that, it has higher classification accuracy with lower computation time. This is because of the hybridization of Fuzzy logic with neural network. Since Fuzzy logic is more suitable for handling uncertainty and Neural networks are good at real time operation, online adaption and efficient this approach has the advantages of both systems. However there are so many advantages, FMNN generates a common problem called overlapping. In the existing papers this can be solved by introducing compensatory neurons in neural network but this will improve the complexity of the system. Also in some existing research works, an improved FMNN is implemented by adding some rules for overlapping to detect possible overlapping areas. This will cause more number of hyper boxes and an additional pruning operation is required, because when there is less number of hyper boxes, the process will be faster. Therefore, a new method is required which will improve the FMNN.

VI. CONCLUSION
As fault tolerance is a huge challenge in cloud data centers, it is very difficult to devise a mechanism to predict the fault in advance. The proposed bi level FMNN-KNN classifier overcomes the drawback present in the previous research work such as classification of unpredictable/overlapping data which is resulted from the FMNN classifier A Fuzzy min max Neural Network classification approach is proposed to predict fault that can anticipate failure in advance. It is proved that prediction to vm failure is possible and demonstrated the results using cloudsim. The results demonstrate that the proposed approach yields better results in prediction of vm failure in cloud data centers.

VII. REFERENCES