A Survey on Failure Prediction Techniques in Cloud Computing

Vaibhavi Prajapati and Vidisha Thakkar

September 3, 2020

A Survey on Failure Prediction Techniques in Cloud Computing

Vaibhavi Prajapati
Information Technology Department
L.D. College of Engineering
Ahmedabad, India
vaibhavi0071@gmail.com

Vidisha Thakkar
Information Technology Department
L.D. College of Engineering
Ahmedabad, India
vidisha.apit2018@gmail.com

***Abstract*— The reliability of cloud service is a crucial issue for cloud service providers and the service users due to large scale demand of cloud computing services for the hosting applications at industrial level. There are two types of fault tolerance techniques namely reactive and proactive to address the issue of reliability. For the systems which require high availability of Virtual Machines the latter technique i.e. Proactive technique is used to make timely actions to reduce any losses. In Cloud Computing, using proactive fault tolerance techniques, issues such as Virtual Machine migration, Virtual Machine placement, Load balancing etc. can be addressed. In this survey paper, various methods are surveyed which predicts the failure in Cloud Systems and improves the reliability of Cloud Environment.**

Keywords*— *Cloud Computing Reliability, Virtual Machine Failure, Failure Prediction, Linear Regression, SVM, ANN, KNN, MING, LSTM, Random Forest

I. Introduction

Cloud computing provides on-demand computing services such as applications, storage and processing power, typically over the internet and on a pay-as-you-go scheme. The services of cloud from cloud service provider can be rented rather than owning his / her own computing infrastructure or data centers.

Below are the three Cloud Service Models: [1]

SaaS:

SaaS or Software as a Service is a model that provides services of software and is available via a third party over the internet. No installations or downloads are required on your existing system with SAAS. Google G Suite, Dropbox, Microsoft Office 365, etc are common examples of SAAS.

IaaS:

Infrastructure as a Service or IaaS provides the provision of virtual computing resources on the cloud such as storage, networking hardware with continuous maintenance and support. Without installing hardware on their premises, businesses can opt for computing resources as per their requirement. The leading IaaS cloud service providers are Amazon Web Services, Microsoft Azure, and Google Compute Engine.

PaaS:

Platform as a Service or PaaS is a cloud-base where the user can develop, test and organize the different applications as per his business. PaaS simplifies the process to develop enterprise software. PAAS provides the virtual runtime environment and gives a favorable platform for developing and testing applications. Examples of PaaS are Google App Engine and AWS Elastic Beanstalk are. PaaS is subscription based i.e. it gives flexible pricing options as per business requirements.

II. Background

Reliability of Cloud systems lies when the cloud services are provided without any interruption. The cloud services are interrupted when there occurs a failure. Failure in cloud computing is as an event that occurs when there is deviation from intended service to delivered service. The failure in cloud systems can be one of the following. [2]

Network Faults: These are the faults that occur due to link failure or partition in network or loss of packet or packet corruption, destination failure, etc.

Physical Faults: These are the faults that occur in hardware. They can be fault in memory or fault in CPU and storage, fault in memory, etc.

Media Faults: These are the faults that occur when the media head crashes.

Processor Faults: These are the faults that occur in the processor because of crashes in the operating system.

Process Faults: These are the faults that occur when there is shortage of resources or there are software bugs, etc.

Service Expiry Faults: These are the faults that occur due to expiration of cloud services for the resource opted while using the application.

III. Literature Survey

To make the cloud computing environment reliable, one way is to predict the failure before it occurs and hence, we can make necessary steps such as VM migration or VM Load Balancing to avoid major loss. Failure prediction is also required for predictive maintenance as it has the ability to prevent failure incidents and thus, maintenance costs. Predictive maintenance anticipates failures and helps to tale proactive actions. Many authors have used different machine learning algorithms to predict the failure. Basically, the prediction problem is classification problem. It results whether the current state of Cloud can result int failure or not. Based on historical pattern found that results in failure, future failures can be predicted in advance and major loss can be avoided. Below are various techniques to predict the failure in the Cloud based environment.

A. Linear Regression and Support Vector Machine [3]

Hussaini Adamu [3] in his paper, compared two machine learning techniques to predict the failure in cloud system. The prediction is based on: Disk, DIMM, OS, Platform, HSV, CPU, and Others. The categories are based on these 7 groups. Two machine learning algorithms are used namely, Linear Regression Model (LRM) and Support Vector Machine (SVM) with a Gaussian kernel. Time is taken as the predictor variable and hardware failures are taken as the response variables. A linear regression model (LRM), takes into account the relationship between the predictor and response variable and is linear in nature. They obtained the prediction results using Linear Regression, SVM with Gaussian Kernel, and SVM with Polynomial Kernel. The results show that component failures and time shows correlation.

In the CPU failure group, the failures in the year 2006 was over 90, in 2007 the it decreased to over 40, and in 2008 a light decrease was there above 20. The DIMM failure group shows that in 2006 the number of failures was over 350, in 2007 there was a drastic drop to about 150 and in 2008 it decreased to about 100. The paper shows that the SVM outperforms the Linear Regression Model. The predicted result got by using these machine learning algorithms, shows that there exists a correlation between failure of components and time. That is, as time increases, failure rate decreases.

However, as the number of predicted year's increases, both models (especially the linear model) fails, thus consistently gives a zero value which is because of insufficient data obtained. Better results can be obtained if the data on which prediction is done spanned over many years. A failure prediction model should focus on accuracy and also on how easily the predicted result can be interpreted. Also, the prediction is not performed on large real time data of cloud environment.

B. *Message Pattern Learning [4]*

Purvil Bambharolia [4], defined a technique that consists of two different modules in his paper. One module predicts the failure using message logs that are passed between various components of cloud. The other module detects the failure after prediction. The proposed architecture for failure prediction is made up of these 3 phases:

(a) pre-process phase: it makes classification of input messages & then finds patterns of failure. In this phase, message patterns are derived from message logs that are recorded early. This phase has three steps: message classification, failure information extraction and generating message patterns. In message classification, the message gets divided into: timestamp, priority, message source, message text. Then the messages are classified on the basis of their similarity index. Thus, two messages fall into same group if they share many common words. In failure information extraction, the messages that are classified as Error, Critical, Alert or Emergency are considered that shows failure occurrence. The failure information is extracted once the message classification results as a failure. In message patterns generation, message patterns are generated for prediction. A message pattern is a set of types of message present in the message window. It can a sequence of messages by either taking into account their order or ignoring their order.

(b) learning phase: it finds the probability of pattern that relates with failure. From the observed message pattern, the probability that a failure occurs in a certain period is estimated using Bayes' theorem using the message pattern dictionary. The probability is then updated in the message pattern dictionary.

(c) prediction phase: it compares the real time messages with the learned messages and predicts the possibility of failure. Failures are predicted by making usage of message patterns and probability that is in the pattern dictionary. With the input message pattern, the reference from the pattern dictionary is taken and the column that matches the input message pattern is taken. Then, calculations of probability for each type of message failure is done. If the probability for any message type of failure is greater than the threshold, it is taken as a sign of failure.

By improving the accuracy of the prediction model, the proposed technique can be enhanced. Advanced prediction models like ANN, recurrent neural networks, etc. can be used to improve the accuracy of the failure prediction. Also, this method can be further used to predict the downtime using prediction analysis.

C. *ARIMA Model [5]*

Ajay Rawat [5] in his paper, used the ARIMA model to predict the failure of VMs. Using this model, prediction of VM failure is done on non-stationary failure trace. ARIMA i.e. auto-regressive moving average is applicable to both stationary as well as non-stationary data. ARIMA is denoted by (p, d, q) where p, d, q $\geqslant 0$ & p, d, q $\in \mathbb{Z}$. p is the order of autoregressive model, d is the degree of differencing, and q is the order of the MA model. For time-series of data, ARMA (p', d') model is given by following equation where $\alpha_0 = -1$, $\theta_0 = 1$, L is the lag operator, θ_i are the parameters of the MA part and α_i are the parameters of the autoregressive part of the model.

Box-Jenkins method is used to predict from historical time series data, the failure using a Failure Predictor Module (FPM). Non-stationary characteristics of Virtual machine failure is taken into consideration. The health info of 3000 VMs is used and observations are made from the weekly patterns of failure collected for 13 weeks. Using the Failure Predictor Module (FPM), the best-fitted ARIMA is found that has a minimum Bayesian information criterion (BIC) values & Akaike information criterion (AIC). Conversion of the non-stationary time series data to stationary data is done by log-transformed data's d order differencing. The Root mean square error (RMSE) is seen as 0.0457, Mean absolute error (MAE) as 0.0345, Mean absolute scaled error (MASE) as 0.6036. ACF and PACF plots are plotted to verify the residuals of ARIMA process for testing the accuracy of the prediction. No spikes were seen outside the relevant zone and the autocorrelation values tend to zero for both ACF and PACF plots. This shows that remaining residuals are random. Therefore, accurate predictions are done using the FPM model.

One of the disadvantages of the ARIMA model is that it becomes unstable, both when changes in observations are made and when changes in model specification are made and is costly because it requires a large amount of data, and there is lacking convenient updating procedures and the fact that it must be estimated using nonlinear estimation procedures.

D. *K-Nearest Neighbor, Artificial Neural Network, Random Forest and Support Vector Machine [6]*

Hussaini Adamu [6], in his paper, used four machine learning algorithms to develop a prediction-based model, that are: Support Vector Machines (SVM), K-Nearest Neighbours (KNN), Artificial Neural Network (ANN), Random Forest

(RF) to identify the overloading or underloading of the resources of hosts.

K-nearest neighbor (KNN) is a supervised machine learning model that uses training data to make classification of unknown instances. These instances are the known class labels and are grouped on their similar properties. KNN is a non-parametric method used to classify and its output is a class membership for different class groups (labels). Artificial Neural Network (ANN) of multiple layers have been used: an input layer, 2 hidden layers and an output layer. Four input neurons are taken in the input layer, A = [A1, A2, A3, A4], three output neurons are taken in output layer, B = [B1, B2, B3] and each hidden layer has 3 hidden neurons, H = [H1, H2, H3]. The neurons at each layer are connected to those of next layer. They are associated with a weight w_i which is calculated while training. The resultant output b_j is calculated with each training point ai and w_i. The network weights are changed using the learning rate equal to 0.4 and momentum equal to 0.2. Random Forest (RF) is used to generate trees and categorize a new object from an input vector by penetrating the input vector on every tree in the forest. Each tree has a unit vote from the input vector that classifies them and the classification that has the maximum votes of all the trees in the forest is selected by the forests. Support Vector Machine (SVM) is a Machine Learning algorithm used by many of authors for prediction. Since, the complexity of SVM is less, it can accurately categorize the data which is linearly separable and is also used for the decision boundaries that are complex in nature. The hyperplanes are identified based on the classes that are grouped by SVM.

Their experimental results are compared that predicts the load using the above-mentioned models i.e. KNN, ANN, Random Forest, and SVM. The comparison results show that the Random Forest model achieves maximum accuracy as compared to SVM, KNN, and ANN. After Random Forest, ANN stood second in terms of maximum accuracy.

Although, reliability of Cloud Systems can be improved by combining these prediction approaches with fault-tolerant techniques. The approach is not tested on real data.

E. AdaBoost Hidden Markov model [7]

Zhixin Li [7], in his paper, used the AdaBoost-Hidden Markov Model as a VM failure prediction method and improved the reliability of VMs which eventually, improved the overall performance of the cloud system. Using AdaBoost influenced Hidden Markov Model, the relationship between the observation state and the hidden state of the Virtual Machine is found and the VM failure state is predicted. This method can be used with any cloud system for predicting the failure state of VMs, which eventually, improves the predictive ability of VM security state.

The security state of VM is difficult to predict that impact the potential security of the VM system. The proposed method makes use of HMM and solves this problem. The prediction of the observed state is done by AdaBoost algorithm. AdaBoost is an iterative algorithm and is used to combine multiple weak learners of linear regression to produce a strong learner. In HMM there is a hidden state and an observation state. The state transition probability is represented by A and observation probability is represented by B, and there is also an initial state π. Parameter λ is set as (A, B, π). The collected observation state data of VMs is used by AdaBoost algorithm to predict the observation state of VM at the next step. Then, preprocessing of the observation state data is done to match the input of the HMM model. The parameter λ is set by the parameter learning in HMM model. Then after, prediction of the present state of VM is done and thereafter, the probability result of the failure of VM is obtained, and it can be known whether the VM is failed or not. The results show that the probability output of HMM is highly influenced by the value of 'c'. The smaller the value of c is, the more accurately it predicts the failure of VM.

However, there is a need in exploration of the relationship between the internal security state and the observation state of VM in the learning phase and prediction process. Also, the learning and prediction process can be further studied improved.

F. MING: Long Short-Term Memory & Random Forest [8]

A node failure is caused by a variety of reasons and is reflected by many temporal and spatial signals. Qingwei Lin [8], in his paper, proposed a novel failure prediction model MING, that uses: a LSTM model to make predictions from the temporal data, a Random Forest model to make predictions from spatial data; a ranking model that ranks the nodes by their failure-proneness.

The temporal features are the attributes that represent a node's state in time (for example, IO throughput, resource usage, response delays, sensor values, etc.). Then can even be collected from the original sources (such as system event counts, log event counts, error or exception event counts, etc.). Spatial features define the global relationships among nodes in the form of explicit/implicit dependency. Deployment segment, load balance group, rack location, and update domain are some of the Spatial features.

For temporal features, LSTM (Long Short-Term Memory) is used, and is a widely adopted deep neural network (DNN) model. It can balance between retaining the previous state and memorizing new information. LSTM captures the patterns behind the time-series data in a better manner and has proven to be successful in solving tasks such as machine translation and speech recognition. Bi-directional LSTM is used for the time series data x_1, x_2..., x_n; x_i is input vector of all temporal features on time i. The LSTM layer generates representation sequence h_1, h_2, ..., h_n, that is passed to a fully connected dense layer. The output of this dense layer is 128 x 1 vector Vt. This vector is passed to a soft-max function, that is final layer of the DNN-based classifier. For spatial features, Random Forest technique is used, and is one of the most used classification techniques. It builds many decision trees during the training time and results the class of the voting result from the individual trees. Random Forest does splitting of the trees based on information gain; hence, it can better reflect the impact of discrete values. Here, an ensemble of total 128 trees are trained. The results h_1, h_2..., h_n are concatenated into a vector V_s and fed to a majority voting module which results into the final classification result.

When both of these features are used, MING is found to achieve an average F1-measure equal to 75.2%. When only the temporal features are used by predicting using the LSTM model, the average F1-measure can be seen to drop from 75.2% to 48.8%. When only the spatial features are used by making predictions using only the Random Forest model, the average F1-measure is seen to drop gradually from 75.2% to 57.1%. we also compare MING with the baseline approaches that are implemented using conventional classifiers including

Logistic Regression, Random Forest, LSTM, and SVM. Also, the improvement in F1-measure of MING model over Logistic Regression, Random Forest, SVM and LSTM is 17.4%, 21.7%, 13.3%, and 14.6% respectively.

However, they have made trade-offs between precision and recall and gave more weightage to precision than recall. Also, LSTM is a bit more time consuming to train the model. Tradeoff has been made with the time consumption of model training.

IV. Literature Review Summary

Following is the table representation of the techniques to predict the failures in cloud systems as discussed above. The authors of the papers surveyed are mentioned along with that paper's research gap.

TABLE 1 SUMMARY OF FAILURE PREDICTION TECHNIQUES

Author	**Aim**	**Algorithm used**	**Research Gap**
Hussaini Adamu, Bashir Mohammed, Ali Bukar Maina, Andrea Cullen [3]	To predict hardware failures in a real-time cloud environment to improve system availability.	Linear Regression Model, SVM	Prediction is based on a smaller number of features. Accuracy can be improved by other ML algorithms.
Purvil Bambharolia, Prajeet Bhavsar, Vivek Prasad [4]	To solve proactive management of failures by failure prediction and detection	Message pattern learning, Bayes' theorem	Accuracy can be improved by using ANN, RNN. Time consuming.
Ajay Rawat, Rama Sushil, Amit Agarwal & Afzal Sikander [5]	The prediction of virtual machine failure based on the time series stochastic model	ARIMA	Based on non-correlated data. Not used real cloud environment
Anju Bala, Inderveer Chana [6]	Aims at analyzing the problem of load prediction by comparing the performance of various ML algorithms.	ANN, SVM, KNN, Random Forest	Accuracy can be improved by combination of various ML techniques.
Zhixin Li1, Lei Liu, Degang Kong [7]	To improve the reliability of VMs and overall performance of cloud platforms by focusing on VM hidden security state	Hidden Markov Model	Observation state of VM is valued only by load, Relationship between observation state of VM and internal security state needs to be explored.
Qingwei Lin, Ken Hsieh [8]	Predicts the failure of a node in cloud system based on historical data.	MING: LSTM + Random forest	Hyperparameters for both models can be optimized, LSTM takes more time

V. Conclusion

In this paper, a survey of various prediction techniques for Failure Prediction in the cloud environment is done. After evaluating well known techniques, it is clear which methods should be used in which kind of environment. Random forest can be used in place of LR and SVM. ARIMA model can be used where there is stationary data. Ada-boost Hidden Marcov Model can be used to predict the hidden security state of Virtual Machines. Of all the methods seen here, LSTM is seen to be the best model to predict the failures in the cloud environment. Though, the training is time consuming as compared to other models. Also, the idea behind the partition of data based on temporal and spatial features leads to better prediction of failure in the cloud systems. With arrival of new techniques, the papers are referred here and it is seen that latest techniques can perform better than old techniques.

Acknowledgment

I would like to extend my sincere thanks to great many people who helped and supported me during the completion of this report. Specifically, I would like to express my sincere gratitude to my internal guide Prof. Vidisha Thakkar for supporting, guiding and rectifying all my actions, steps and documents with attention and care. She has presented decent amount of attention during whole tenure and made necessary corrections as and when needed.

References

[1] M. U. Bokhari, Y. K. Tamandani and Q. M. Shallal, "Cloud computing service models: A comparative study," 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, pp. 890-895, 2016.

[2] G. Singh, S. Kaur, "Review on Fault Tolerance Techniques in Cloud Computing", International Journal of Engineering and Management Research, vol. 7, pp. 69-71, 2017.

[3] A. Bala, I. Chana, "Prediction-based proactive load balancing approach through VM", Journal of Engineering with Computers, Springer-Verlag London, UK, vol. 32, 2016.

[4] P. Bambharolia, P. Bhavsar, V. Prasad, "Failure Prediction and Detection in Cloud Datacenters", International journal of scientific and technology research, vol 6, 2017.

[5] A. Rawat, R. Sushil, A. Agarwal and Afzal, "A New Approach for VM Failure Prediction using Stochastic Model in Cloud", IETE Journal of research 2018.

[6] H. Adamu, B. Mohammed, A. B. Maina, A. Cullen, H. Ugail, I. Awan, "An approach to failure prediction in a cloud-based environment", IEEE 5th International Conference on Future of Internet of Things & Cloud, 2017.

[7] Z. Li, L. Liu, D. Kong, "VM Failure Prediction Method Based on AdaBoost-Hidden Markov Model", IEEE International Conference on Intelligent Transportation, Big Data and Smart City (ICITBS), 2019.

[8] Q. Lin, K. Hsieh, Y. Dang, H. Zhang, K. Sui, Y. Xu, J. Lou, C. Li, Y. Wu, R. Yao, M. Chintalapati, D. Zhang, "Predicting Node Failure in Cloud Systems", ESEC/FSE, Lake Buena Vista, FL, USA, Association for Computing Machinery (ACM), 2018.