# A Virtual Server QoS Enhancement Method in Cloud Computing

Berihun Fekade
Korea University, Department of Computer and
Information Science
2511 Sejong-ro, Sejong City 339-770, S. Korea
+82-044-860-1755
berihunf@korea.ac.kr

Taras Maksymyuk
Korea University, Department of Computer and
Information Science
2511 Sejong-ro, Sejong City 339-770, S. Korea
+82-044-860-1755
taras.maksymyuk.ua@ieee.org

Minho Jo
Korea University, Department of Computer and
Information Science
2511 Sejong-ro, Sejong City 339-770, S. Korea
+82-044-860-1348
minhojo@korea.ac.kr

## ABSTRACT

Virtualization is one of the most promising solutions to create a flexible and powerful technology for cloud computing paradigm. Newly upcoming technologies such as Internet of Things (IoT) and cloud computing have brought so many new cloud-connected devices, resulting in the increased load of cloud computing. In such conditions, system reliability becomes a challenging task for cloud computing. In this paper, we propose a failure prediction and prevention model of hypervisors by using Bayes naive classifier. The advantage of this model is to give higher accuracy of prediction and early failure detection by exploiting real-time sensed data using past experience.

## Categories and Subject Descriptors

G.3 [**Probability and statistics**]: *Probabilistic algorithms, Distribution functions, Time series analysis.*

## General Terms

Algorithms, Design.

## Keywords

Cloud computing; hypervisor clustering; failures prediction;

## 1. INTRODUCTION

Emerging paradigms such as cloud computing [1] and Internet of Things [2] have opened a wide scale of new opportunities for industry and casual human life lately. These emerging technologies have a stable computational power on remote physical servers in the cloud. Individual separate cloud consumers share the same cloud resources such as computational power, storage, and network infrastructure. In such a case, load of physical resources will be non-uniform and highly unpredictable because of many requests from separate consumers with different requirements in terms of processing time and complexity. This will cause higher number of server failures. Thus, lower reliability of entire cloud infrastructure should be solved and we propose a new failure prediction and prevention model by clustering hypervisors and using Bayes naive classifier with past experience. The major contribution of the paper is to have a cloud environment that has an algorithm which learns from past experiences of failure. The algorithm collects past failure data points from each cluster of hypervisor and arranges them in their sequence. Using this collected points, our algorithm checks real time physical server attributes to know how much they resemble with those previously collected data points. Using Bayes naïve classifier we get the probability of resemblance with each of the collected data points.

We choose Bayes naïve classifier since it is good for few variables, simple to use, and it compute the multiplication of independent distributions in the data [6]. And it has shown a higher performance in comparison with complex neural networks and decision tree [9]. Failure prediction and prevention in cloud computing gives the cloud environment to have a stable and active infrastructure.

## 2. RELATED WORK AND PROPOSED SYSTEM

In this paper, we propose a new heuristic prediction model based on groups of attributes, including CPU load, RAM load, and load of consumer requests according to the past time before failure. One physical server has one corresponding hypervisor. The hypervisor is responsible for creating and hosting multiple virtual servers inside a physical server. In our model, each hypervisor knows the real time status of its underlying physical server, for example, CPU load, RAM load, data input/output rate, and number of consumer requests. In addition, the hypervisor provides virtual servers with access to resource pools of the physical server. When a physical server fails, the corresponding hypervisors also consequently fail.

Physical servers in a cloud environment tends to fail due to many factors. The failure conditions other than physical damage on the physical server and power disconnection can be monitored through their individual hypervisor's virtual infrastructure manager (VIM). Early detection of physical server or hypervisor failure before it happens plays a vital role in QoS of the cloud environment. It will make the cloud to be available for users without interruption.

TinyChecker provides transparent detection and recovery of virtual servers against hypervisor failures [5]. To discover

IMCOM '16, January 04-06, 2016, Danang, Viet Nam
© 2016 ACM. ISBN 978-1-4503-4142-4/16/01…$15.00
DOI: http://dx.doi.org/10.1145/2857546.2857629

failures, TinyChecker identifies the context of special-purposed hypervisor actions. The context can be a clue or background of the actions. This hypervisor for TinyChecker applies nested virtualization which is the act of running a hypervisor nested within another hypervisor. While TinyChecker can only detect failure of hypervisor and recover specific hypervisor from failure, our proposed mechanism maintains even system reliability. It migrates the virtual servers in advance by analyzing the behavior of the hypervisor inside the cluster before failure occurs.

In order to satisfy the quality of experience (QoE), it is very important to keep the minimal allowable number of hypervisors always alive. Thus, we are motivated to propose an effective idea to minimize the number of hypervisor failures in this paper. We group hypervisors in order for fail safe mechanism before we apply Bayesian classifier. Our proposed system is based on grouping hypervisors for fail safe mechanism.

As the number of requests increases inside a cluster of hypervisors, the VIM gathers attributes of each individual physical server by using heartbeat messages [3]. The heartbeat messages show the physical attribute values. Fig. 1 illustrates the scenario. When a hypervisor fails, the virtual server belonging to the failed hypervisor can be migrated to another hypervisor (or equivalently meaning another physical server). So, one or more hypervisor(s) should exist in advance inside the cluster for migration in case any hypervisor fails. The data from the failed hypervisor can be moved into the other hypervisors. The data from the failed hypervisor are distributed among several hypervisors in parallel to achieve faster data migration.

A hypervisor cluster [4] is a group of independent hypervisors with its corresponding physical servers working together as a single system to provide high availability of services for consumers. When a failure occurs on one hypervisor in a cluster, resources are redirected to and the workload is redistributed to another hypervisor inside the same cluster, respectively. The cluster mechanism gives a higher level of availability, reliability, and scalability comparing to a single hypervisor. The main goal of clustering hypervisors in cloud computing is to make each one of them highly available even in a situation when there is a tendency for failure. Higher availability of cloud resources is a crucial issue for cloud consumer’s prospect.

We use the Bayes naive classifier [6] [7] to predict expected time of failure according to the current status of the physical server by knowing historical data from previous failures. The previous failure data are collected to the several groups according to the interval time record before any failure occurs. For instance, if a hypervisor fails at time t, our prediction model checks average value of attributes of each server at time t-2 minutes, two-step ahead time and group it to the “2 minutes before failure” for that individual hypervisor. The average value of attributes between 2 and 5 minutes before failure will be grouped to “5 minutes before failure”. In the same way, other failure interval times for the existing data will be grouped so that the status of attributes will be checked in real-time under these groups. These groups will provide failure prediction interval time.

At the moment when the predicted failure interval time is known for each hypervisor, the VIM (Virtual Infrastructure Manager), shown in Fig. 1, will check previous predicted time before failure to check whether it is increasing or decreasing. If the predicted prediction shows a continuous decrease pattern, the hypervisor will be considered as possible failure. In such a case, the VIM will choose another hypervisor with lowest tendency to fail for clustering for alternative solution, i.e., for migration of data and resources. After that, migration of virtual servers between hypervisors can be performed. This re-clustering operation of virtual servers inside each cluster allows to minimize the number of failures and provide stable cloud performance. This is the main advantage of our proposed work in this paper.

The proposed method is shown in the following steps to give a more detail view. The assumptions on our model is that, there is a cluster of hypervisors with previous log files that shows the physical server attributes values. By analyzing the log file our model chooses the failure points. The purpose of the algorithm is to give the highest probable failure group for a real-time physical data point attributes.

Step 1. Pin point previous failure points in each cluster of hypervisor.

Step 2. Select data points that occur before each failure points.

Step 3. Group each data points that are selected at step 2 in relative to the failure points that are collected in step 1. For instance 1 minute from the failure point, 2 minutes from failure points and so on. This failure groups will be assigned for each data points and we will use them as failure groups.

Step 4. Using normal distribution models for each attribute (for instance CPU load or RAM load) in one failure group that is also in each cluster we find its mean and variance.

Step 5. Using the mean and variance we calculate the probability of our target real-time data attribute (i.e. the one that we want to know its resemblance with those failure data points) for each failure groups inside one cluster.

Step 6. After getting the probability of all the real-time attributes in accordance with each failure group, we use Bayes naïve to calculate the probability each failure group given that the real-time data is given.

Step 7. After comparing each group probability, we select the maximum probability result to set the real-time data.

Step 8. When the failure group of the real-time data becomes near to the lowest failure group (for instance 1 minutes to failure group), this gives us time to rearrange the data inside the specific physical server into a more stable physical server(the one that has a more stable hypervisor failure group prediction).

## 3. EXPERIMENTAL RESULTS

For our experimental test, we used one of the servers of MonALISA repository for ALICE [8]. Physical server attributes for one month period were collected using the FZK server farm of ALICE. Collected server attributes from the FZK server are CPU usage, RAM usage, CPU IO wait time, and TCP/UDP packets. We assumed FZK server will be on failure when the server traffic reaches more than 1.0GB/sec. The red line in Fig. 2 shows the point where we assumed failure will occur. From this point we arranged failure groups to give us a view of how the server behaves when it becomes failure. From one month collected data, we obtained forty four occurrences of server traffic above 1.0GB/sec.

Using the forty four occurrences we made the failure groups before the server traffic reached the 1.0GB/sec point. We named group names starting from 1 which is the one that occurs near to the 1.0GB/sec point. We collected a set of groups within 1 minute interval and averaged them together to give a group of failure in accordance with the 1.0GB/sec occurrence moment. We made a range of groups of failure from 1 minute to failure to 180 minutes from the forty four points and obtained training and test data sets. We selected 20% of the data points at random from the collected data as test sets and 80% as training sets, we checked their failure group using our model.

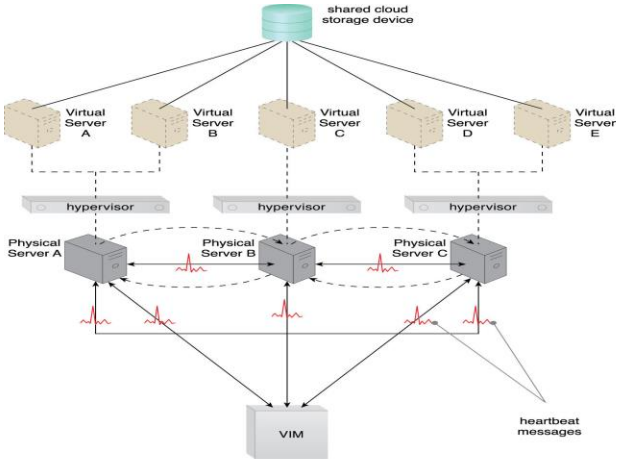

**Figure 1. A cloud architecture resulting from the application of the hypervisor clustering pattern [3].**

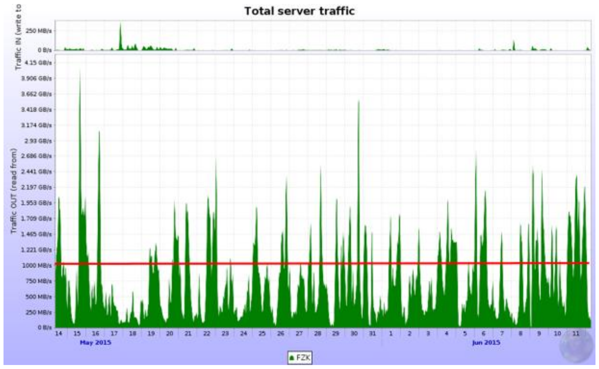

**Figure 2. MonALISA repository server traffic.**

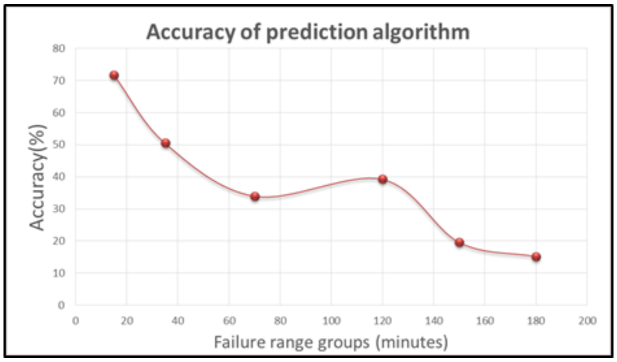

**Figure 3. Accuracy of prediction algorithm**

The experiment gives 71.78% accuracy of prediction of their assigned failure group within 15 minutes failure group range. As we widen the range into 35 minutes, the prediction of their assigned failure range accuracy becomes 50.53%, and we continue to increase the range into 70 minutes, 120 minutes, 150 minutes, 180 minutes and we get 34%, 39.25%, 19.63% and 15.22% respectively. The result is shown in figure 3 above.

## 4. CONCLUSION

The prediction accuracy shows a high accuracy around the range the failure occurred, but when we increase the range of failure groups, the accuracy becomes low. The reason for this result is that the data point attribute values becomes similar and they don't show a significant change as they were near to the selected failure point. We will continue to do more experimental work using enough training data. Thus we expect to obtain higher prediction accuracy within the range of failure point. We introduce on-going work with the idea of "hypervisors clustering" which has never been done before in cloud computing. In a virtualized cloud environment, clustering hypervisors is beneficiary in such a way that if one of the hypervisor fails, active virtual servers can be transferred to stable one.

## 5. ACKNOWLEDGEMENT

This work was funded by the National Research Foundation (NRF) and the Ministry of Education (MOE) of Korea under the BK21 Plus project.

## 6. REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, and M. Zaharia. "A view of cloud computing", *Communications of the ACM*, vol. 53, no.4, pp. 50-58, 2010. DOI = http://dl.acm.org/citation.cfm?id=1721672

[2] J. Gubbia, R. Buyyab, S. Marusic, M. PalaniswamiJ, "Internet of Things (IoT): A vision, architectural elements, and future directions", *Future Generation Computer Systems*, no.29, pp. 1645-1660, 2013.

[3] T. Erl, Z. Mahmood, R. Puttini, "Cloud Computing: Concepts, Technology & Architecture", *Prentice Hall Service Technology Series*, US, 2014.

[4] V. Chavan, P.R. Kaveri, "Clustered virtual machines for higher availability of resources with improved scalability in cloud computing", *IEEE First International Conference on Networks & Soft Computing*, pp. 221-225, Aug. 2014.

[5] C. Tan, Y. Xia, H. Chen, B. Zang, "TinyChecker: Transparent Protection of VMs against Hypervisor Failures with Nested Virtualization", *IEEE/IFIP 42nd International Conference on Dependable Systems and Networks Workshops (DSN-W'2012)*, pp.1-6, June, 2012.

[6] Y. Tang, W. Pan, H. Li, Y. Xu, "Fuzzy Naive Bayes classifier based on fuzzy clustering", *IEEE International Conference on Systems, Man and Cybernetics*, vol.5, p.6, Oct. 2002.

[7] A. Kelemen, H Zhou, P. Lawhead, Y. Liang, "Naive Bayesian classifier for microarray data", *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 3, pp. 1769 - 1773, July, 2003.

[8] http://alimonitor.cern.ch/display?page=xrdagg/lan_wan

[9] T. M. Mitchell, Machine Learning, *The McGraw-Hill Companies, Inc.*, 1997.