ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Auto Detection and Diagnosis of Faults in Web Application over Cloud Computing

Kshitija S. Nandgaonkar, Swarupa Kamble

ME Student, Dept. of Computer Engg, RMD Sinhagad School of Engg, Pune, India

Assistant Professor, Dept. of Computer Engg, RMD Sinhagad School of Engg, Pune, India

ABSTRACT: Cloud computing has been widely used platform for ecommerce application. In modern era cloud computing facing many challenges for fault diagnosis in application. Due to dynamic nature of cloud computing, application's workload and resource allocation fluctuates dynamically which causes behavior of web application to change to unexpected condition. In large scale dynamic environment it becomes very difficult for manual operators to find out and fix the root cause of failure. To overcome these issues, this paper proposes a framework that will automatically detect and diagnose the faults in cloud based web application. In this paper we recognize access behavior pattern and also correlate its workload with performance/resource utilization metrics. The framework identifies abnormal metrics which causes system behavior to change and automatically resolve faults to bring an application to healthy status.

KEYWORDS: Cloud computing; fault dignosis; web application; correlation analysis; monitoring tool.

I. INTRODUCTION

In recent years, many web commerce application are deploying over public cloud environment to offer internet based services to customer. The development and deployment of many cloud based web application are prone to many types of faults because of dynamic cloud environment. These vulnerability of application to many types of faults may lead to application failures which results in economic loss. A large population of users and also corporate companies are affected by these failures. For example, in august 2013, amazon.com was down due to an unexpected fault which results in economic loss of million. According to the report of tellme network [5], detecting a faults accounts for 75% of failure recovery time where as detecting faults in time could prevent 65% of failures. Thus, for internet-based services fault diagnosis is necessary for ensuring reliability and performance of application. However, debugging and testing of web application cannot eliminate the irresistible faults triggered at runtime.

Fault diagnosis technologies have been widely used in recent years for web applications. For example, there are some commercial monitoring tools like IBM Tivoli, HP OpenView and Amazon CloudWatch that allows the system operator to manually set rules for a particular system metrics. Whenever value of these metrics exceeds predefined threshold, it will automatically raise an alerts. But manual setting of suitable threshold for number of metrics is extremely difficult in cloud based web application.

In large-scale dynamic cloud computing environment, there are many reasons that makes faults detection very difficult [7]. First, cloud computing actually based on virtualization techniques like VM resizing, VM migration, Vm cloning etc that offers on-demand and flexible services to users. With such dynamic environment, application model changes frequently with dynamic workloads and resource allocation. This will create a challenge for modeling application status and identifies their abnormal behavior from normal behavior. Second reason that makes fault detection difficult is due to large number of metrics collected from different layers.

II. RELATED WORK

H. Chen, G. Jiang, K. Yoshihira, and A. Saxena [2] has presented a solution to accommodate past experiences for the failure diagnosis. They established a knowledge base that stores the information about past failures including their

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

causes, symptoms, recovery strategies, and so on. However in case of dynamic cloud environment, manual setting of these threshold values is very difficult.

Mike Y. Chen, Anthony Accardi, Emre Kcman, Jim Lloyd, Dave Patterson, Armando Fox, Eric Brewer [3] has presented a runtime paths for managing failures in large, distributed systems. If application performance deviates from expected result, faults are detected. However this method can detect faults at application level and unable to detect faults caused due to resource contention.

T.Wang, W.Zhang, J.Wei, and H.Zhong [4] introduces incremental k-mean method for clustering workloads and used a local outlier factor (LOF) for anomalies detection in each specific workload pattern. But the computation complexity of LOF increases with the increase in data instances. Also it can only detect faults but unable to locate their root cause. These limitations are overcome by using system described in our paper.

L.K. Joshila Grace, V.Maheswari , Dhinaharan Nagamalai [6] analyzes web logs of user. These logs files are maintain for client as well as web server. By using the error logs file, our approach can resolve fault occurring at application code level. Thus we can manage the form upload, file access like methods to be made secure and auto recover.

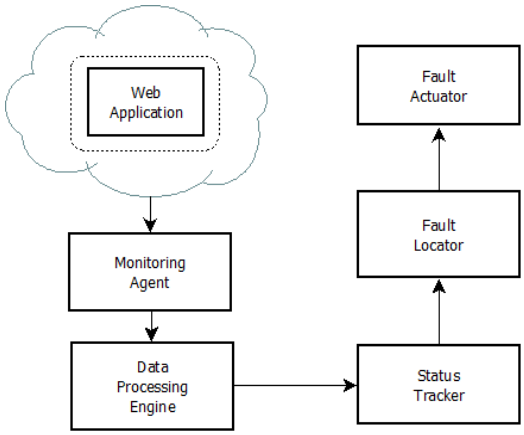
A. *Problem Statement :*

To develop a framework that will automatically detect faults occurring in web application over cloud platform and find out the root cause of failure to resolve the faults to bring application to healthy status. The proposed framework will automatically diagnose known types of faults at application level as well as at code level.

III. PROPOSED METHODOLOGY

A. *System Architecture:*

Fig. 1 shows proposed framework for automatic detection and diagnosis of faults occurring in web application over cloud system. The framework includes five components as monitoring agent, data processing engine, status tracker, fault locator and fault actuator.



- Monitoring Agent : Monitoring agent provide us the logs which will contain the measuring parameters like CPU utilization, memory utilization etc. Monitoring agent collects access pattern of customers and pass this information to next component i.e. Data processing Engine.
- Data Processing Engine : This component performs two important task. First it collects data from monitoring agent and second task is removing duplicate entries and constructing different clusters of access pattern according to access pattern of customers. Here main target is to group similar access behavior into an access behavior pattern.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

- Status Tracker: Since the number of users influences the performance of application, we need some way to model correlation between these two parameters. For this, status tracker uses canonical correlation analysis (CCA) to determine correlation between workload and performance utilization [9].
- Fault Locator: The fault locator aims at finding the metrics which more likely causes the application behavior to change from normal behavior.
- Fault Actuator: Once we know the root cause of failure, this component uses some VM technique to relieve the application from failures. Also by analyzing error logs file of web page, we can manage code level faults.

B. *Description of the Proposed Algorithm:*

This section describes how the proposed framework will be implemented. The data processing engine of framework characterizes access behavior of application and recognize customers' access pattern by grouping it into different clusters according to their similarity [8]. Algorithm 1 describes density based clustering [1] method for clustering access pattern of customers.

Algorithm 1 Density Peak Clustering

Input : Data Instance Set *S*
Output : Cluster Label Set *L*

1. For each data point *i* in *S*
   1.1 For each data point *j* ($j \neq i$)
      1.1.1 Compute distance *dij* between i and j
2. For each data point *i*
   2.1 Compute local density ld_i of point *i*
   2.2 Compute the minimum distance with higher density md_i
      2.2.1 If point *i* is in higher density
         2.2.1.1. $md_i = \max(d_{ij})$;
      2.2.2 Else
         2.2.2.1 $md_i = \min(d_{ij})$;
3. For each point *i*
   3.1 T-transfer m_i (ld_i and md_i) to fit *t*-distribution:

$$t(m_i) = \frac{m_i - \overline{m_i}}{S_i} \sim t(L-1)$$

Where $\overline{m_i} = \frac{1}{n}\sum_{j=1}^{L} m_{ij}$, n is number of data instances $S_i = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(m_{ij} - \overline{m_i})^2}$

4. Calculate the candidates *i* of cluster centers:
   $m_i > (\overline{m_i} + S_i)$
5. Calculate cluster centers.
6. For each point *i*
   6.1 If ($md_i > \overline{m_i} + S_i$ and $ld_i < \overline{m_i} - S_i$)
      6.1.1 Point *i* is regarded as an outlier
   6.2 Else
      6.2.1 Point *i* is assigned to same cluster

The important task after fault detection is localize the metrics in which fault occurs. The problem of fault localization is addressed by Feature selection method.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Feature selection method involves two approaches filter method and wrapper method. The filter method scrutinizes many uncritical features from the metrics and the method is known as ReliefF [1]. This method removes redundant features by searching two nearest neighbors, one from same class called as nearest hit and other from different class called as nearest miss [10].

Algorithm 2 ReliefF – Problem Localization

Input : metric set *M* ; positive instance set *P,* negative instance set *N*
Output : metric weight set *W* [*M*]

1.Set all weights $W[M] = 0$;

2. For $i = 1$ *to* m do

2.1 Randomly select an instance R.

2.2 Find k-nearest hit $h_1, h_2, \ldots . h_k$;

2.3 Find k-nearest miss $m_1, m_2, \ldots . m_k$;

2.4 For $i = 1$ to all_ metrics do

2.4.1 $W[M] = W[M] - \frac{1}{k}\sum_{i=1}^{k}(dist(M,R,h_i) - dist(M,R,m_i))$;

In second phase, we use wrapper method to rank features selected in ReliefF. The wrapper method uses SVM-RFE [1] to train feature set to build the classifier. This method also identifies workload deviation from the existing one and provides the result to the actuator.

IV. RESULTS AND DISCUSSIONS

We have developed private cloud platform on which our fault diagnosis framework resides. We deployed one simple web application on this platform to validate fault diagnosis framework. We evaluate how well fault diagnosis framework can recognize status of web application, detect the injected faults, locate the root causes of fault and correct the fault.

The experiment environment consist of four VMs , each VM having a memory of 110 MB. We deploy web application over this private cloud. An Apache Tomcat provides deployment environment for the application. A backend tier, Mysql database server is used.

We select some servlets of application to inject faults at random. We are dealing with four types of faults:

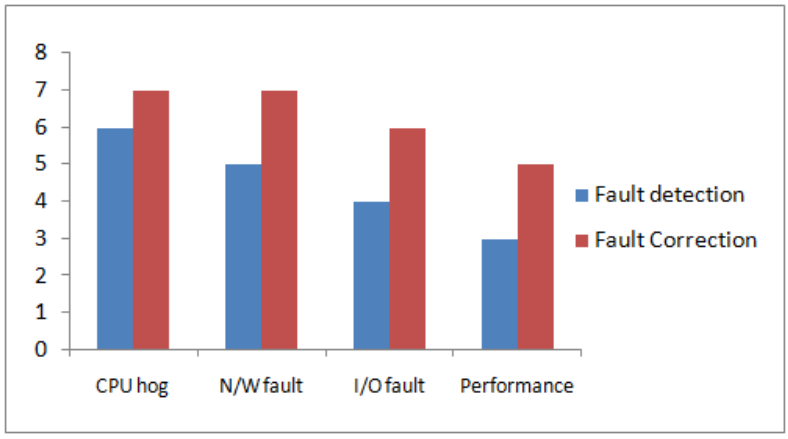
1. Network Fault : This type of fault leads to deny of services because of service saturation. When application server is down, then network fault occurred.
2. I/O Fault : I/O fault related to I/O operation. This kind of fault can be triggered, when we stop database service.
3. Internal Fault : Internal faults includes CPU hog, which results from endless loop or circular wait in source code. We inject this fault in application by inserting additional computation operations occupying CPU circles.
4. Performance Anomaly : This is caused by increasing workloads. This fault is injected by increasing maximum number of user logged into web application. For instance, number of user at time is limited to certain value, if user exceeds this value, then this type of fault is occurred.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016



V. CONCLUSION AND FUTURE WORK

In this paper, we propose a framework that automatically detect and resolve faults in web application deployed over cloud computing environment. The framework uses a clustering method to identify access pattern of user and correlate workloads with performance/resource utilization. Fault localization is carried out by using feature selection method. The proposed method tries to resolve detected faults automatically at application as well as at code level. Currently, our approach does not monitor application components. In future work, we will monitor application components to locate faulty components.

REFERENCES

1. Tao Wang, Wenbo Zhang, Chunyang Ye, Jun Wei, Hua Zhong, Tao Huang, " FD4C: Automatic fault diagnosis framework for web applications in cloud computing, in IEEE Transaction on systems MAN and Cybernetics: Systems, 2015
2. H. Chen, G. Jiang, K. Yoshihira, and A. Saxena, Invariants based failure diagnosis in distributed computing systems, in *Proc*. 29th IEEE Symp. Rel Distrib. Syst., New Delhi, India, 2010, pp. 160166.
3. M. Y. Chen et al., Path-based failure and evolution management, in *Proc*. 1st Symp. Netw. Syst. Design Implement., Berkeley, CA, USA, 2004, pp. 2336.
4. T. Wang, W. Zhang, J. Wei, and H. Zhong, "Workload-aware online anomaly detection in enterprise applications with local outlier factor,"in *Proc. 36th IEEE* Annu. Comput. Softw. Appl. Conf., Izmir, Turkey, 2012, pp. 25–34.
5. D. Oppenheimer, A. Ganapathi, and D. A. Patterson, "Why do Internet services fail, and what can be done about it?" in Proc. 4th Symp. Internet Technol. Syst., Seattle, WA, USA, 2003, pp. 1–16.
6. L.K. Joshila Grace, V.Maheswari , Dhinaharan Nagamalai, "Analysis of web logs and web user in web mining" in International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.1, January 2011, pp.99-110
7. G. Jiang, H. Chen, and K. Yoshihira, "Modeling and tracking of transaction flow dynamics for fault detection in complex systems," IEEE Trans. Depend. Secure Comput., vol. 3, no. 4, pp. 312–326, Oct./Dec. 2006.
8. A. Rodriguez and A. Laio, Clustering by fast search and find of density peaks ,Science, vol. 344, pp. 14921496, Jun. 2014.
9. D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural Comput., vol. 16, pp. 26392664, Dec. 2004.
10. K. Javed, H. A. Babri, and M. Saeed, "Feature selection based on class-dependent densities for high-dimensional binary data," IEEE Trans.Knowl. Data Eng., vol. 24, no. 3, pp. 465–477, Mar. 2012.

BIOGRAPHY

Kshitija Nandgaonkar received B.Tech. degree in Information Technology in 2013 from Government College of Engineering Amravati(An Autonomous Institute of Government of Maharashtra) and pursuing M.E. from RMDSSOE, Warje, Pune.

Swarupa Kamble is working with RMDSSOE, Warje, Pune as an Assistant Professor. She has experience of 5 yrs in the field of teaching and research.