Black Box Anomaly Detection in Multi-Cloud Environment

Mahendra Kumar Ahirwar
PG-Scholar
UIT-RGPV, Bhopal (India)

Manish Kumar Ahirwar
Asst. Professor
UIT-RGPV, Bhopal (India)

Uday Chourasia
Asst. Professor
UIT-RGPV, Bhopal (India)

ABSTRACT

Automatic identification of anomalies for performance diagnosis in the cloud computing is a fundamental and challenging issue. TPA is interested to identifies these anomalies and remove them so that the performance of the cloud systems increased. In this paper we are proposing an Automatic Black Box Anomaly Detector which can find anomalies automatically with minimum human intervention. Using this detector we can find old and even new anomalies created in the cloud computing systems even if we don't have knowledge of source code (i.e. black box testing). Automatic black box anomaly detection is a two step process in which first of all data from different sources is collected and transform it into a common form that is act as input for black box anomaly detector and secondly anomaly detection is performed.

General Terms

Anomaly detection in cloud computing.

Keywords

Black box anomaly detector, cloud service provider, performance diagnosis, cloud systems.

1. INTRODUCTION

Cloud computing is an Internet-based most recent popular technology offering dynamic resources, scalable resources, on-demand, self-service and pay-per-use. Cloud computing is an active area for research and growing very fast. It provides services at low cost and low operational software and hardware expenditure's. The use of cloud computing has increased in companies rapidly because of fast access to applications and decreasing maintenance cost for cloud infrastructure.

A. Types of clouds

Clouds can be classified into four categories on the basis of physical location of users. Sumit [15] provides these types of clouds to user for their business according to their necessity. These clouds types are private, public, community and hybrid clouds. In the available types clouds he explain benefits and limitations of each cloud types on the basis of which we can conclude that which cloud model will be suitable for us. A *private* cloud is one which is setup by single organization and installed services on its own data center. *Public* cloud services are offered by third-party cloud service providers and involve resource provisioning outside of the user's premises. The *Community* cloud can offer services to the cluster of organizations. In other words we can say that community cloud provides combinational services of a group of clouds. *Hybrid* cloud is the combination of any two or more than two types of clouds which are mentioned above means combine any two or more from private, public or community to build it.

B. Cloud service models

Cloud computing technology allows users to access information and computing resources from anywhere that a network connection is available. It provides a shared pool of services and resources including data centers (data storage space), networks (Internet), computer processing power and user applications. Web server provides services from shared pool according to 3-tier architecture. Pankaj [14] says that services are provoded by the providers can be divided into three types which are explained as:-

Infrastructure-as-a-Service (IaaS):- It is used to provide network for connecting users and servers and also provides virtual machines to start, stop, access and configure virtual servers and storage blocks. *Platform-as-a-Service (PaaS)*:- In this model a platform is provided to users which typically include operating system, programming languages, execution environments, databases, queues and web servers. *Software-as-a-Service (SaaS)*:- This model provides "On-demand software's" to users without installation setup and running of the applications.

The remaining parts of this paper are organized as follows:- Section 2 describes related work and section 3 includes open research issues followed by problem identification in section 4 for our work. In section 5 we provide our proposed methodology to resolve mentioned challenges of section 3 now section 6 shows architecture of black box anomaly detector which perform two phase diagnosing process to remove anomalies available in the tracing data. Section 7 is consisting of experimental setup and result and Finally in section 8 we conclude our paper and also provide direction for future enhancement.

2. RELATED WORK

2.1 Motivation Towards Multi-cloud Environment

The terms "multi-clouds" or "interclouds" or "cloud-of-clouds" that were introduced by Vukolic. These terms suggest that cloud computing paradigm should not be ended with a single cloud. Using their illustration, a cloudy sky consolidated different colors and shapes of clouds which provide directions to different implementations and administrative domains. Recent research has focused on the multi-cloud environment which control multiple clouds and avoids dependency on only single cloud. Cachin et al. identify two layers in the multi-cloud environmental architecture: the base or first layer is the inner-cloud while the upper or second layer is the inter-cloud. In the upper layer, the Byzantine fault tolerance finds its place. Now we will first summarize the previous work on Byzantine protocols over the last three decades.

2.2 Anomaly Detection in Cloud Computing

There are three types of anomaly detection techniques which are available for cloud computing. These anomaly detection techniques are: - Statistical, data mining and machine learning.

i) Statistical anomaly detection

In this technique of anomaly detection, to identify anomalies system observes computations and generate a profile which stores a value to represent their behavior. For anomaly identification this technique used two profiles in which one is stored ideal profile and other is current profile which is updated periodically and calculates anomaly score. If anomaly score of current profile is higher than
Threshold value of stored profile than it is considered as anomaly and then it can be detected.

ii) Data mining based anomaly detection

Anomalies can also detected using data mining techniques like classification, clustering and association rule mining. To identify anomalies this technique added level of focus to anomaly detection. Data mining techniques used an analyst which can differentiate normal and abnormal activity within clouds by defining some boundaries for valid activities in the clouds.

iii) Machine learning based anomaly detection

This approach of anomaly detection uses the concept of machine learning to identify anomaly. The ability of programs or softwares to learn and improve performance of the task or group of tasks over time is called machine learning. This technique develops a system which can improve performance of the programs on the basis of previous results. From the previous results new information is acquired and on the basis of this information even execution strategy can be changed for performance improvement if required.

Statistical anomaly detection technique is beneficial as compared to other two techniques because this technique have number of benefits over others. Firstly this technique does not require any prior knowledge domain of security risks or intrusion. Secondly this technique has capability to detect even very recent anomalies generated in the data. This also provides accurate notification for anomalies that occurs over extended time period.

This paper is the extension of the [1] in which CloudDiag was used to detect anomalies and these anomalies are diagnosing. Now in this paper we have used another approach for tracing of data without knowledge domain i.e. Black box anomaly detection in which collection, assembling, transformation and monitoring of tracing data is performed and then detection and diagnosis of anomalies is done. This paper is surveyed as:-

CloudDiag [1] can diagnosis the anomalies which are appeared in case of fine granularity, unsupervised and scalable cloud systems. CloudDiag can perform, performance diagnosis in three steps i.e. collection of data, assembling of data and diagnosis of anomalies. Diagnosis of anomalies is to find out anomalies in the tracing data and if present then remove these anomalies. Anomalies diagnosis process is done as identify anomalous categories, identify anomalous methods and locate physical node where anomalies are found. CloudDiag can also have scalability property to identify anomalies available in any other neighbor cloud of distributed system.

Automatic black box anomaly detection [11] can detect anomalies if available in the data. The sources from where data is collected may generates data in different form so this data is transformed into a common form of data and then anomaly can be removed. Anomaly detector is a working model for anomalies removal in three phase process. Three phase of anomaly detection are train, test and evaluation. At training phase normal data is used to compute some values for model and these values are decided according to the properties of data. At next stage data is compare with the values of models and if matched then passed to map-reduce matrix. At last stage map-reduce matrix perform some computations and if any deviation (anomaly) is appeared then it will be removed from here.

Dapper [5] introduces an infrastructure for monitoring of performance of services. It stores tracing data into Bigtable [6]. This approach is unable to describe how diagnosis is performed for anomalies. Spectroscope [2] introduces to find primary causes of performance changes between two time intervals. P-Tracer [7] can be able to identify anomalies available in the call trees and once the anomalies are detected these can be removed from the data.

Pinpoint [8] can be used to traces the request call relationship of service components and apply clustering algorithm to classify data entries into failure or success group. From this entries classification we can identify anomalies i.e. entries available in the failure group. In CloudDiag [1] latency-anomalous methods and their corresponding physical replicas are identified.

Pip [9] and Ironmodel [10] compare the actual behavior of data with self-defined expected data to determine whether a request of user is anomalous or not. But it is very hard to design such models because these models require vast amount of specific domain knowledge. In cloudDiag [1] request latencies are considered as intrinsic properties to determine anomalous methods invocations which require no specific domain knowledge.

Lakhina [12] compute entropies of data entries in BigTable and use it for automatic classifying data anomalies through unsupervised learning. LERAD [13] used different approaches to tackle different types of data entries. A framework is discovered that uses both normal and anomalous data to find out characteristic features of anomalies on the basis of which anomalies can be removed. Shobha [11] designed an anomaly detector which uses black box data tracing mechanism to identify anomalies in data if available without any specific domain knowledge.

3. OPEN RESEARCH ISSUES

In cloud computing detection of anomalies is real world challenging task because of service hosting on the cloud servers or data centers. Attackers always try to find out any understandable pattern from the service or data for making alteration in it. In case of multi cloud environment where we distribute services into fragment for security purpose it is very important to detect anomaly if it is attached with any fragment of the service. In the previous year's lot of research is done on anomaly detection in traditional web computing and also on cloud computing. We have done survey on anomaly detection in internet as well as cloud computing in which various framework/models are designed. Every framework or model uses an anomaly detection technique. In table 1 we are providing summary of various anomaly detection framework (or models) with techniques used in it.

Table 1 Comparison study of various anomaly detection techniques

S.No.	Framework/Model	Anomaly Detection Technique	Advantages	Limitations
1	Pinpoint	Clustering data mining technique	Suitable for large and dynamic systems where it is difficult to monitor application-level knowledge of services	Performance degradation of services and issue of scalability
2	LERAD	Unsupervised learning technique	It can detect stimulated and real attacks	Unable to differentiate between true and false alarms
3	Pip	Imperative and declarative statistical tehniques	Monitoring and checking dynamic properties of programs like latency, throughput, concurrency and node failure	Require vast amount of specific domain knowledge
4	Probabilistic likelihood	Bayesian Network Technique	Can detect anomalies in case of categorical datasets	Difficult to define constraints for groups and not suitable for real valued attributes
5	P-Tracer	Supervised learning technique	Simple and easy to implement for anomaly detection	Detect only primary causes of anomalies
6	EbAT	Statistical techinique	Detect anomalies from whole metrics simultaneously instead of individual value of metrics	Neither focus on possible cases of anomalies nor evaluate scalability, cross-stack metrics and hadoop
7	Ensemble of feature chains	Supervised learning technique	Handle numeric and nominal featured data and suitable for mobile devices	It suffers from high false positive rate
8	ADD	Recursive learning technique	Tackle unlabelled data	Unable detect failure for which cloud operators are used
9	Monitoring-as-a-Service	Machine learning and event processing rules	Tackle cross-VM side channel attack and multi-tanency in clouds	Complex computation process for anomaly detection
10	TARA	Scheduling algorithms	Minimize latency of thermal anomalies and maximize accuracy of datacenters	Difficult to detect anomalies in case of high density hotspots
11	CloudDiag	Statistical technique	Diagnose anomalies in fine-grained, scalable and unsupervised cloud systems	Required specific knowledge domain for anomaly detection

4. PROBLEM IDENTIFICATION

At SaaS layer of public cloud, performance diagnosis is challenging task because data is reside at remote locations where security issues and secrete data outsourcing affect the performance of cloud service provisioning. In cloud computing services are provided to users, easily and efficiently. A cloud service is composed of many components that are designed by different teams and even a single component is made up of many replicas (i.e. component instance). These replicas are distributed in the different physical nodes of cloud systems can be assembled into multiple types of services for serving large amount of user requests. Service provisioning in the cloud is also arise some anomalies like Service-level-agreement violations by software faults, unexpected workloads or hardware workloads. These defects may be ignored due to manifested only in a small part of component replica. But cloud computing systems faces so many real-world challenges for applying performance diagnosis in the new design of distributed cloud systems (multicloud) as given:-

1. Detection of novel anomalies:- In the tracing data if any novel anomaly that not yet seen in the data is appeared then it is difficult to find this kind of anomaly because detection techniques are limited to detect only those anomalies which are already present in the samples.
2. Extracting features of interest for anomaly detection:- In the data in which lot of fluctuations are present but only little regularities of normal data are available then it is difficult to extract features of our interest which are important for anomaly detection.
3. Performance diagnosis in fine granularity:- In the cloud computing a component has made up of many replicas and there are so many performance related private and public methods in it. These replicas are distributed in the clouds so it is very challenging to localize anomalous method and corresponding physical location of replicas.
4. Unsupervised performance diagnosis:- A cloud service is composed of many components which are developed by different teams. These teams can be updated independently online so it is very difficult to maintain the behavior models for such evolutional systems.
5. Performance diagnosis with scalability:- In case of multi clouds components and their replicas may be located in the different clouds and they may be defected so in this case diagnosis with scalability is required which is very challenging issue.
6. Performance diagnosis without knowledge domain:- If the source code of services are available then white box tracing mechanism can be applicable for performance diagnosis. But if they are not available then white box

tracing mechanism cannot be applied for performance diagnosis.

7. Handling of diversity of data sources:- Today's numerous and diverse of data sources are available and these sources provides data in different form so handling of these different form is also a challenging task.

Cloud computing systems have above challenges regarding performance diagnosis. Typical cloud systems are service-oriented in nature and the response time of user requests directly reflects the system performance. Recent work in [2, 3] has shown that it is possible to find out performance anomalies with end-to-end request tracing data. However, an efficient and unsupervised diagnosis tool for locating fine-grained performance anomalies is still lacking. This gap can be bridge by CloudDiag [1] tool. Performance diagnosis tool CloudDiag with white-box tracing data mechanism periodically collects the end-to-end data tracing from each physical node of the cloud systems and then employs a customized Map-Reduce algorithm to collect and assemble the tracing data of each user request. Then the tracing data is classified into different categories according to call trees of the requests. When the cloud system is observed performance degradation (i.e. average response time of user request is larger than the threshold value), a cloud operator can trigger CloudDiag tool with its web interfaces for performance diagnosis. With the tracing data of requests, CloudDiag will perform a fast customized matrix recovery algorithm to identify the method calls (together with the replicas they locate) which contribute the most to the performance anomaly. The whole process requires no domain-specific knowledge to the target service it means this tool is unsupervised. So CloudDiag with white-box tracing data mechanism can solve problems associated with [2, 3] but it cannot work if the source codes of the services are not available. In case if source codes of services are not available then instead of white-box, another black-box tracing data mechanism of CloudDiag can be used.

5. PROPOSED METHODOLOGY

When a user request to a service, it may go through many component replicas, and invoking numerous methods they provide. During methods invoking a call tree of a user request is generated which is directed tree describing the method invocation relations. Where each node is a method and each edge $e = u \rightarrow v$ from node u to node v denotes that method v is invoked by method u that is u is a caller and v is its callee. It is also possible that requests for the same service can generate multiple call trees. Multiple call trees are possible because of existing multiple paths for accessing same service for example the call tree of one request reading a file from the cache is different from that of another request reading a file from the disk. To solve above listed problems, in this paper we are proposing an approach of CloudDiag which can resolve all the problems. Our proposed scheme used black box anomaly detector which can detect and remove anomalies presented in the tracing data.

6. BLACK BOX ANOMALY DETECTION

Automatic black box anomaly detection for performance diagnosis in cloud computing systems is two phase process:-

- Collection and transformation of data:- At this step two operations are performed i.e. collection of data and transformation of data. Collection of data means collecting data from the various available sources in which we have to find anomalies. Transformation is used to transform collecting data from various sources to a common format of data which is provided as input to the anomaly detector.
- Detection and diagnosing of anomalies:- At this stage also two operations are performed these are detection of anomalies and diagnosis of anomalies. In the first operation it is to be checked whether anomaly is available or not in the data. If anomaly is present in the data then second operation for diagnosing anomaly is performed.

6.1 Phase-1 Collection & Transformation of data

Two operations which are performed at this phase are described as:-

6.1.1 Collection of data

CloudDiag traces user requests at particular interval to avoid bad impact on the data. Each component replica records the performance data and save them in its local memory from where CloudDaig collects data at regular intervals and check for anomalies if available or generated during computations.

6.1.2 Transformation of data

This operation is performed to transform different forms of data collecting from various sources to a common form of data which will be supplied as input to the anomaly detector.

6.2 Phase-2 Detection and diagnosis of anomalies

This phase also required two phase for removal of anomalies present in tracing data:-

6.2.1 Detection of Anomalies

Anomalies can be detected using BigTable [1] where data collected from each replica is stored. Fast map-reduce matrix can be used to find out anomalous categories on the basis of their distribution latencies because when a service passes through normal and abnormal replicas then distribution latency of service passes through abnormal replica must be affected. A cut-off value is used to identify anomalous category, if distribution latency of any service category is greater than cut-off value then this service category is considered as anomalous. Once anomalous service category is detected we can easily identify anomalous methods and corresponding physical location of replicas using RPCA (Robust Principal Component Analysis) algorithm. RPCA [4] is an fast customized matrix recovery algorithm.

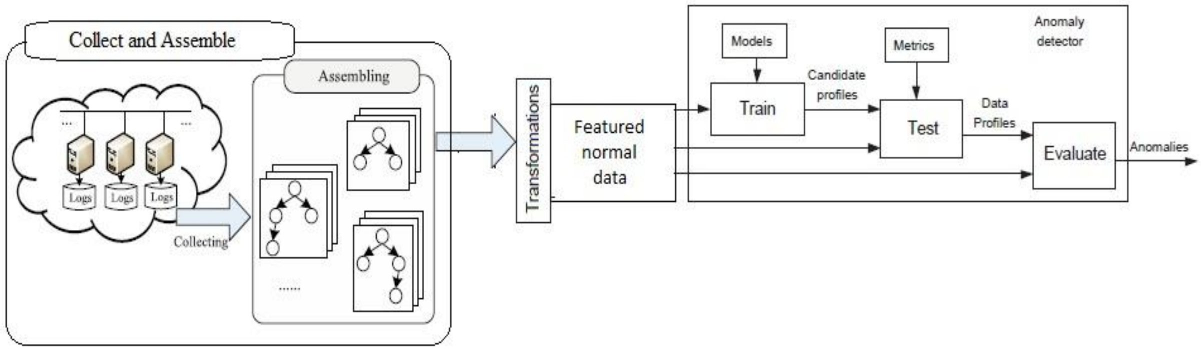

Fig. 1 Architecture of Black Box Anomaly Detector

6.2.2 Diagnosis of Anomalies

Black box anomaly detector performs three operations to diagnose anomalies present in the tracing data. These operations are training, testing and evaluation. Training operation is used to compute values for the models on the basis of normal data. This operation computes value of model on the basis of normal data by learning from the previous computations and examples. Testing operation is used check and selects only those values from the output of training operation which satisfies conditions determined from the normal data. Evaluation is real-time anomaly diagnosing operation which can store only those outputs of training and testing data which don’t have deviations (anomaly) in it and discard rest of data.

7. EXPERIMENTAL SETUP AND RESULT

For anomaly detection in multi-cloud environment we use Cloud Simulator (CloudSim 3.0.3) for starting servers and virtual machines and executed on NetBeans IDE. The services which are delivered to the users are hosted on the cloud servers, provided by CloudBees.com.

Results are generated using the various stored patters of anomalies in the database and an auto-updater is bind up with database to update automatically whenever it receive information about any newly identified anomaly globally. The pattern of identified anomaly is stored in the database and, if it is encountered in the future this anomaly will be detected by black box anomaly detector.

For efficiency consideration, CloudDiag is required to be scalable to the massive performance data. Since CloudDiag conducts the tracing data collection and assembly proactively, the anomaly diagnosing step is the only issue that will influence the scalability of CloudDiag. We study the efficiency of the RPCA-based anomaly detection approach. The inputs are the performance data of a typical category of requests to the SendMail service, which bears a critical call tree that contains 117 methods. There are about 4 million of requests following this call tree each day. Fig. 2 plots the computation time of the anomaly detection approach under different request numbers. It shows that the computational time of the approach also scales almost linearly with the performance data volumes of up to 100 thousand requests. This demonstrates the high scalability of the RPCA-based anomaly detection algorithm. The process of computing a 1,00,000x117 matrix takes less than 200 seconds.

We compare results of CloudDiag approach with our proposed technique and results are represented into tabular form.

Table 2 Comparison of Scalability of existing and proposed anomaly detection technique

S.No.	Computation Time (Seconds)	Request Numbers (10^3)	RPCA Anomaly Detection	Black Box Anomaly Detection
1	20	10	13	15
2	40	20	22	30
3	60	30	41	49
4	80	40	85	88
5	100	50	88	100
6	120	60	115	120
7	140	70	125	140
8	160	80	162	175
9	180	90	178	180
10	200	100	195	196

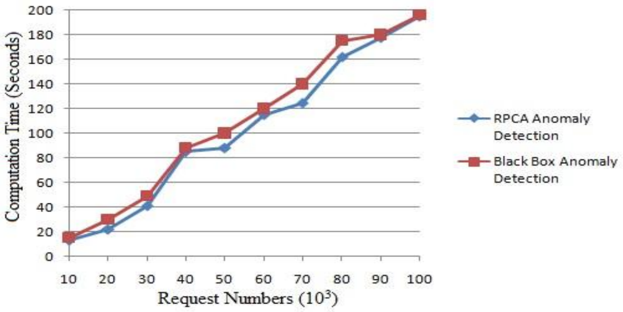

Fig. 2 Scalability of existing and proposed approaches

7.1 Evaluation of Data Splitting

In this section, we demonstrate how CloudDiag helps operators detect real-world performance anomalies that happened in Alibaba cloud computing platform. We adopt the following two measures to evaluate the effectiveness of CloudDiag:-

1. Precision
2. Recall

Precision

The precision is a parameter which measures the exactness of our approach. Precision can be calculated using the formula given below:-

$$Precision = \frac{TP}{TP + FP}$$

Where,

TP = Number of true positives i.e. the number of anomalous methods

FP = Number of false positives i.e. the number of normal methods that are mistaken for the anomalous

Recall

Recall is another parameter which measures the completeness of our system. Recall can be calculated using given below formula:-

$$Recall = \frac{TP}{TP + FN}$$

Where,

TP = Number of true positives i.e. the number of anomalous methods

FN = Number of false negatives i.e. the number of anomalous methods that are mistaken for the normal

To show the advantage of adopting black box anomaly detector in CloudDiag, we compare black box anomaly detector with CloudDiag a recent performance diagnosis approach based on the RPCA algorithm [16]. The approach employs PCA and the Mann-Whitney hypothesis test to identify anomalous methods.

For the map-reduce cluster, one important parameter is the split size, i.e., the volume of data assigned to each Map task. The split size determines the number of Map tasks. A smaller split size indicates that more Map tasks are required to process a given data set. We vary the split sizes from 32 to 512 MB for three data sets (with trace entries sizes being 120, 160, and 200 million lines of trace logs). The computational time of the map-reduce procedure is shown in Table 3 and its graphical representation in Fig. 3. We can see the cluster performs the best when the split size is 128 MB. Hence, in the rest of our experiments, we set the split size 128 MB.

Table 3 Computation time under different volumes of split sizes for CloudDiag white box and black box approach

S. No.	Computation Time (seconds)	Split Size (MB)	CloudDiag White Box Approach			Black Box Approach		
			120 Million	160 Million	200 Million	120 Million	160 Million	200 Million
1	100	512	425	606	863	423	601	859
2	300	256	300	368	447	297	362	443
3	500	128	190	219	288	189	216	283
4	700	64	200	256	304	197	252	299
5	900	32	209	243	328	204	239	326

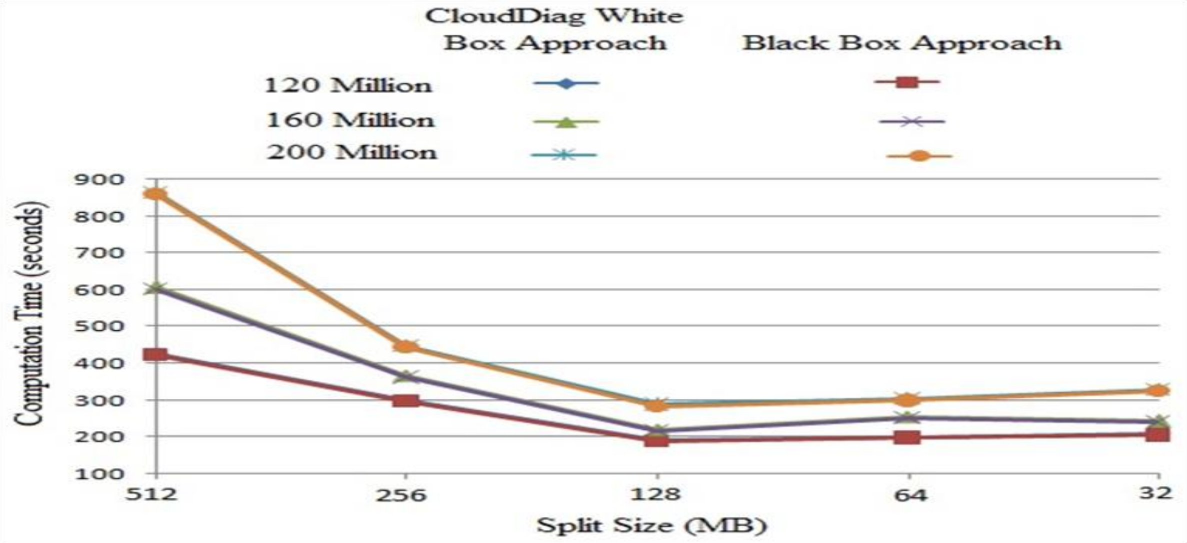

Fig. 3 Computation time under different volumes of split sizes for existing and proposed approach

8. CONCLUSION & FUTURE WORK

Automatic black box anomaly detector for CloudDiag is an anomaly detector which can identify and remove anomalies available in the tracing data of requests. This detector performs anomaly detection process in two phases. First it collects data from various sources (in our case from various component replicas) and transforms this data into a common type of data which can be considered as input to the anomaly detector. At the second phase actually anomaly detection is performed which is also a three step process i.e. training, testing and evaluation. This automatic anomaly detector can also handle massive amount of data using the scalability property of the cloud system.

From the future enhancement point of view, this proposed black box anomaly detector can now able to tackle massive amount of data but this detector can be optimized and then it can be design for data warehouse.

9. REFERENCES

[1] Haibo Mi, Huaimin Wang, Yangfan Zhou, Michael Rung-Tsong Lyu and Hua Cai, “Toward Fine-Grained, Unsupervised, Scalable Performance Diagnosis for Production Cloud Computing Systems.”IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, pp 1245-1254, June-2013. .

[2] R. Sambasivan, A. Zheng, M. De Rosa, E. Krevat, S. Whitman, M. Stroucken, W. Wang, L. Xu, and G. Ganger, “Diagnosing Performance Changes by Comparing Request Flows,” Proc. USENIX Eighth Symposium Networked Systems Design and Implementation (NSDI), pp. 43-56, 2011.

[3] M. Chen, A. Accardi, E. Kiciman, J. Lloyd, D. Patterson, A. Fox, and E. Brewer, “Path-Based Faliure and Evolution Management,” Proc. USENIX Symposium

Networked Systems Design and Implementation (NSDI), pp. 23-36, 2004.

[4] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust Principal Component Analysis?" Arxiv Preprint arXiv:0912.3599, 2009.

[5] B. Sigelman, L. Barroso, M. Burrows, P. Stephenson, M. Plakal, D. Beaver, S. Jaspan, and C. Shanbhag, "Dapper, a Large-Scale Distributed Systems Tracing Infrastructure," Technical Report dapper-2010-1, Google, 2010.

[6] F. Chang, J. Dean, S. Ghemawat, W. Hsieh, D. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber, "Bigtable: A Distributed Storage System for Structured Data," ACM Trans. Computer Systems, vol. 26, no. 2, pp. 1-26, 2008.

[7] H. Mi, H. Wang, Y. Zhou, M.R. Lyu, and H. Cai, "P-tracer: Path-Base Performance Profiling in Cloud Computing Systems," Proc. IEEE 36th Ann. Computer Software Applications Conference (COMPSAC), pp. 509-514, 2012.

[8] M. Chen, E. Kiciman, E. Fratkin, A. Fox, and E. Brewer, "Pinpoint: Problem Determination in Large, Dynamic Internet Services," Proc. IEEE International Conference Dependable Systems and Networks (DSN), pp. 595-604, 2002.

[9] P. Reynolds, C. Killian, J. Wiener, J. Mogul, M. Shah, and A. Vahdat, "Pip: Detecting the Unexpected in Distributed Systems," Proc. USENIX Third Symposium Networked Systems Design and Implementation (NSDI), pp. 115-128, 2006.

[10] E. Thereska and G. Ganger, "Ironmodel: Robust Performance Models in the Wild" ACM SIGMETRICS Performance Evaluation Rev., vol. 36, no. 1, pp. 253-264, 2008.

[11] Shobha Venkataraman, Juan Caballero, Dawn Song, Avrim Blum and Jennifer Yates" Black Box Anomaly Detection" Carnegie Institute of Technology at Research Showcase @ CMU. Pp 127-132, 2006.

[12] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. *ACM SIGCOMM 2005*.

[13] M. V. Mahoney and P. K. Chan. Learning Rules for Anomaly Detection of Hostile Network Traffic. Third IEEE International Conference on Data Mining.

[14] Pankaj Sareen. "Cloud Computing: Types, Architecture, Applications, Concerns, Virtualization and Role of IT Governance in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, pp 533-538, 2013.

[15] Sumit goyal. "Public vs Private vs Hybrid vs Community-Cloud Computing: A Critical Review", I.J. Computer Network and Information Security, pp. 20-29, 2014.

[16] H. Mi, H. Wang, G. Yin, H. Cai, Q. Zhou, and T. Sun, "Performance Problems Diagnosis in Cloud Computing Systems by Mining Request Trace Logs," Proc. IEEE Network Operations and Management Symp. (NOMS), pp. 893-899, 2012.