Enhancing Performance of a Standalone System from Event Logs using K–medoid Algorithm

Mario Pinto & Vineet Pukhraj Jain

Department of Computer Engineering, Goa College of Engineering, Goa University
E-mail : mariopinto87@gmail.com, vineet.bumb@gmail.com

***Abstract* – Event logs deals with monitoring of system performance. Data collected by performance log helps in analyzing the behavior of the standalone system in terms of memory used and CPU time taken by each service running on the system. Performance of a system is evaluated by analyzing the resources used by each service. Clustering the services based on the memory usage and CPU time helps the system administrator in tracking the services causing system to slowdown. In system troubleshooting, performance logs helps the administrator to catch problems earlier by analyzing the abnormality in the trace. Many of the Apriori-based algorithms [1] takes longer time to do the analysis when dealing with huge amount of performance log data. Furthermore, many other event classification techniques cannot analyze the performance degradation of the system automatically. This paper focuses on automatic analysis of the performance of a system, by tracking the services causing the system to slowdown using the performance logs. To get the clusters of services from performance logs, the algorithm used is the K-medoid. K-medoid algorithm forms the clusters of services based on the percentage of memory and CPU time used by each service. This approach helps in detecting and destroying the services causing the system to slowdown such that the performance of the system is enhanced. The paper shows the proposed method and the experimental results obtained on the generated performance log trace.**

Keywords – Clustering, Data Mining, K-medoid, WEKA, Performance Logs

I. INTRODUCTION

Event logging and monitoring provides resourceful information about the state of a system. Performance logs analysis helps in knowing the resource utilization by the services running on the system. In event correlation a set of events that takes place in a given time interval is interpreted and then processed for the task of fault management in a system. Event monitoring through most of the event correlation tools takes place with the help of algorithms like the APRIORI [1] algorithm which are sometimes inefficient in correlating longer events.

Mining patterns from event logs helps to detect frequent patterns from log files, to build log file profiles, and to identify anomalous log file lines [4]. The aim of this work is to develop a visualization tool that can be used to view performance log files based on the clusters produced. It helps in easing the summarization of vast amount of data contained in the log files. The contents of event logs helps in indicating the status of the system [8].This makes them indispensable in systems administration and network management and are used by administrators in their general monitoring tasks, security analysis and also for trouble shooting when downtimes occur. Automating events enables administrator to detect problems earlier and improve system performance.

The basic idea in this project is to design a system that can automatically track and manage the performance of a standalone system by analyzing the performance logs of services that are running on the system by using K – medoid clustering algorithm. Here the focus is given on tracking the performance logs of a standalone system such that by tracking the events the administrator comes to know what activity has caused the problem in [2] the system. It tries to find why the system performs slower when certain services run infinitely or consumes more memory. By analyzing these services the administrator can analyze the situation and take the corrective step to get the system back on the correct track.

The paper is organized as follows. Section 2 deals with related work done on event logs. In Section 3, the related concepts of event logs and K–medoid algorithm are presented. In Section 4, the proposed method to

carry out the analysis required for the proposed algorithm is given. In Section 5, the experimental results obtained are presented. Section 6 concludes the paper.

II. RELATED WORK

Considerable amount of work is carried out in the field of event logs. [1] discusses about the log file data sets where in the distance between two points is measured using Euclidean distance formula. [1] also talks about some algorithms which cannot help in discovering patterns from log files.

Event classification in log audit involves sorting events according to priorities and categorizing audited events to cover various systems' functions [5]. To accomplish its function, an auditing system runs in a privileged mode to oversee and monitor all operations. The audit log view included parameters like date and time stamp, database name, type of action, the class of that action, status of the logged action, user credential details, etc, while each row of the audit view includes an instance of an event captured by the audit log system. This work does not help in doing the analysis of performance degradation of the system. It only protects the system from unauthorized access. It could not track the activity causing the system to fail abnormally. Event logs have been recognized among the few mechanisms for gaining visibility into the behavior of a system. The current logging mechanism cannot analyze system performance through event logs. Traditional analysis techniques cannot do proper correlation among entries in the log and hence they cannot do the proper analysis [7]. In [5], any log parameters could be configured for any information flow system.

In [7], Event logs represent a valuable source of data to conduct a failure analysis. Here logs data is used for gaining visibility into the behavior of a system. The paper further discusses about how the traditional techniques might underestimate failure analysis in log files because of the faults in correlating the log entries.

Logging Mechanism

The logging mechanism does not report all the events occurring in the system. Unreported events decrease the level of trust on log-based analysis. Hence it reduces the effectiveness of corrective actions performed by administrator, and lead to wrong insights into the behavior of the system.

Logging mechanism lacks the reporting ability observed for the logs such that when certain services fail to be completed on time, they does not get tracked so that the performance of the system cannot be achieved completely.

Logging mechanism deals with three activities such as collection of event logs, filtering and analysis of the log entries. Collection of log entries involves detection of events that are generated at runtime. Filtering technique involves removing of un-wanted data. Analysis involves around statistical and graphical representation of the activities taking place in the system [7].

In [9], a framework for generating system events from raw textual logs has been discussed. Here representative message set is used to represent information about all the messages occurring such that each message represent one type of event. The approach splits the entire log into different time frames. Although the framework made use of K-medoid algorithm to get the clusters, it was not automated and it took lot of time to do the analysis.

III. BASIC CONCEPTS

A. *Log Trace*

A log is a record of the events occurring within a system. Logs are composed of log entries, each entry contains information related to a specific event that has occurred within a system. Logs contain the information related to system performance activities [3]

Level	Date and Time	Source	Event ID	Task Category
Information	1/10/2013 1:00:38 AM	MSSQLSSQLEXPRESS	17403	(2)
Information	1/10/2013 12:08:29 AM	MSSQLSSQLEXPRESS	17896	(2)
Information	1/9/2013 11:08:29 PM	MSSQLSSQLEXPRESS	17896	(2)
Information	1/9/2013 11:06:18 PM	Windows Error Reporting	1001	None
Information	1/9/2013 10:18:24 PM	VSS	8224	None
Warning	1/9/2013 10:15:24 PM	EventSystem	4609	Event Service
Warning	1/9/2013 10:15:24 PM	EventSystem	4609	Event Service
Warning	1/9/2013 10:15:24 PM	EventSystem	4609	Event Service
Warning	1/9/2013 10:15:24 PM	EventSystem	4609	Event Service
Warning	1/9/2013 10:15:24 PM	EventSystem	4609	Event Service
Warning	1/9/2013 10:15:24 PM	EventSystem	4609	Event Service
Warning	1/9/2013 10:11:48 PM	Avira Antivirus	4113	Infection
Warning	1/9/2013 10:11:48 PM	Avira Antivirus	4113	Infection

Fig. 1 : Example of application windows event log trace

B. *Performance Monitoring*

Performance is the measure of how quickly a computer completes application and system tasks. Overall system performance might be limited by the access speed of the physical hard disks, the amount of memory available to all running processes, the top speed of the processor, or the maximum throughput of the network interfaces.

C. K-medoid Algorithm

K-medoid algorithm is used to get the clusters of services. The dataset obtained from the performance log is given as a input to get the cluster of services, such that the services are clubbed into different clusters based on their memory and CPU time usage. K-medoid algorithm calculates the distance between two services based on Euclidean distance measure. The services having similar memory and CPU time are clubbed into one cluster.

D. Formula

Euclidean Distance Method to calculate the distance between two points: x and y

$$d_p(x,y) = \sqrt[p]{\sum_{i=1}^{n} | x_i - y_i |^P}$$

IV. PROPOSED METHOD

The proposed system's objective is to automatically track and manage the performance of a standalone system by analyzing the performance logs of services that are running on the system by using K–medoid clustering algorithm. The algorithm gives the best clusters of services as a output based on the resources used by each service running on the system. Based on the clusters formed, the system detects the services taking more resources and destroys them such that they does not cause system to degrade in performance because of resource usage by those services. The design of the proposed system is as shown in figure 2 below:

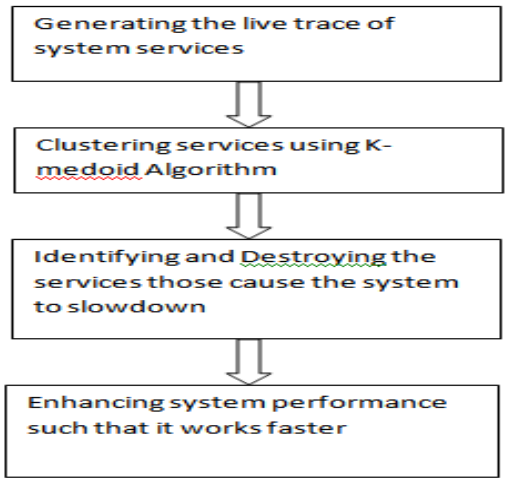

Fig. 2 : Design of the proposed system

The paper is proposed to find the causes behind the slowing down of the system due to the services running on the system by analyzing the performance event logs from the system. Performance log of services are checked with respect to memory usage and CPU time. The generated log trace from the system is given as dataset input for the k-medoid algorithm to get the clusters of services taking more memory and CPU time usage. The paper focuses on the evaluation of resources utilization by the services and it attempts to destroy those services which takes more resources or causes other services to slowdown [7]

Generating the trace of services

Live trace of services (system and user) running on the system are generated with respect to the memory usage and CPU time taken by each service. System services required to run the system and the services started by the user are tracked in the log in CSV format. The trace generated shows the percentage of memory and CPU time taken. The trace generates the log of events for the services which are using considerable amount of memory rest services are discarded.

Clustering of services using K-medoid Algorithm

The dataset that is generated is given as an input to the K-medoid algorithm to form the clusters of services. The services taking more memory and CPU time are clubbed in one cluster and the services taking less resources are clubbed in different clusters. Based on the resource utilization, analysis on the services are made which might cause the system to slowdown.

Enhancing System Performance

The services taking more resources and causing the system to slowdown are analyzed and destroyed such that they do not run infinitely causing other services to stop.

The proposed system automatically gives the clusters of services and tries to destroy the services taking more resources such that the system does not slowdown and the performance of the system is achieved [6]

WEKA tool for Analysis

Clusters of services formed by K-medoid are analyzed and visualized using WEKA tool. It shows the theoretical reports and graphical outputs generated by the K-medoid algorithm with respect to the trace generated. The CSV file generated by the trace is given as a input WEKA.

V. EXPERIMENTAL RESULTS

A. *Performance Log Trace Generation*

Live traces of services running on the system are tracked using Python with respect to memory and CPU time usage. The trace shows the percentage of memory used and CPU time used by each service in CSV format as shown:

Fig. 3 : Performance log trace

B. *Cluster Output generated*

The performance log trace generated from the system is given as a dataset input to the K-medoid algorithm which gives the clusters of services based on the resources utilization as shown:

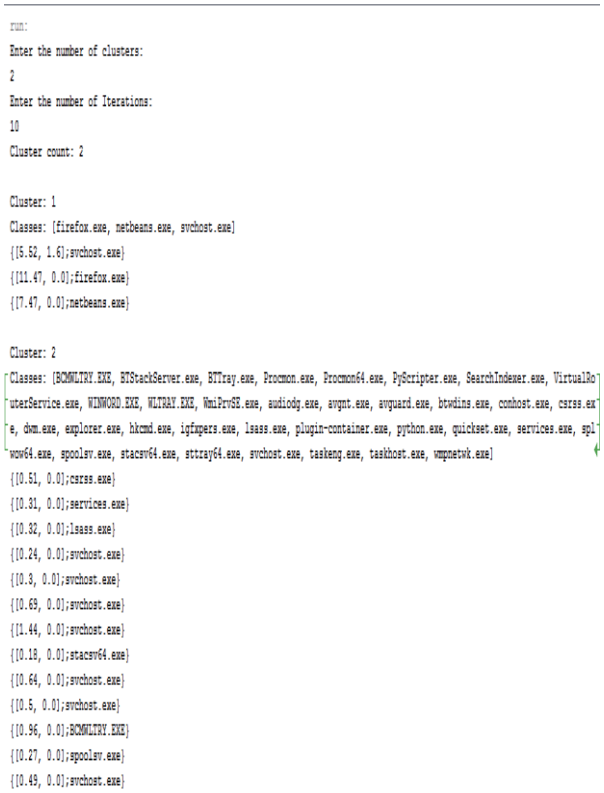

Fig. 4 : Cluster output of services

The above output shows the cluster of services formed based on their memory and CPU time usage. Further After getting the clusters, system performance is achieved by destroying the services priority wise such that the services which takes more resources and which are not important can be destroyed. By doing this, the performance of the system can be enhanced by allowing the services that are most important and which takes less amount of memory and CPU time usage.

VI. CONCLUSION

Event logs are an important tool in system monitoring. Regular auditing log file data allows potential incidents to be reported as they occur and may also assist in preventing future occurrences. By tracking and analyzing the performance logs of a system, the administrator can come to know about the resources that are used by the services. K-medoid clustering algorithm being more robust generates the clusters of services based on their memory and CPU time usage. The clusters having the services with maximum resource utilization are analyzed and destroyed such that system does not slowdown and the performance of the system is achieved by killing the unwanted services.

VII. ACKNOWLEDGMENT

This work was performed as part of a ME thesis in System Performance Monitoring from Event Logs using K-medoid Algorithm. We want to acknowledge the contribution of our colleagues from Goa Engineering College for all the support that was provided.

VIII. REFERENCES

[1] Risto Vaarandi, " A Clustering Algorithm for Logfile Data Sets", Copyright ©2003 IEEE, Reprinted from Proceedings of the 2003 IEEE Workshop on IP Operations and Management (ISBN: 0-7803-8199-8)

[2] Adetokunbo Makanju, Stephen Brooks, A. Nur Zincir-Heywood, Evangelos E. Milios, " LogView: Visualizing Event Log Clusters"

[3] Karen Kent, Murugiah Souppaya, "Guide to Computer Security Log Management", Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, Special Publication 800-92, September 2006

[4] Risto vaarandi, "Tools and Techniques for Event Log Analysis"

[5] Sabah Al-Fedaghi and Fahad Mahdi, "Events Classification in Log Audit", International Journal of Network Security & Its Applications (IJNSA), Volume 2, Number 2, April 2010

[6] Shalini S. Singh and N. C. Chauhan, "K-means v/s K-medoids: A Comparative Study"

[7] Antonio Pecchia ,"The Use of Event Logs for the Analysis of System Failures", Federico II University of Naples via Claudio 21, 80125 – Napoli, Italy November 2011.

[8] Best Practices: Event Log Management for Security and Compliance Initiatives in the European Union, By Ipswitch, Inc. Network Managment Division

[9] Liang Tang and Tao Li, "LogTree: A Framework for Generating System Events from Raw Textual Logs"

◈◈◈