ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Monitoring and Analysis of Integrated System for Application Service Managers

Prerana S. Mohod, K. P. Wagh, Dr. P. N. Chatur

M.Tech Student, Dept. of Computer Science and Engineering, GCOEA, Amravati, Maharashtra, India

Assistant Professor, Dept. of Information Technology, GCOEA, Amravati, Maharashtra, India

Associate Professor, Dept. of Computer Science and Engineering, GCOEA, Amravati, Maharashtra, India

ABSTRACT: With the recent virtualization technologies, in today's data centre share physical or virtual resources of application. Hence it takes a lot of time to monitor system health and historical data of application service managers. In these studies on integrated monitoring software suggest that it shortens the time required for monitoring of collecting and processing of historical data with five basic data formats. The results of survey show that proposed monitoring software with all of the basic data formats supports 98.16% of monitoring tasks. This result suggests that the integrated monitoring software is used effectively for reducing the cost of monitoring software and enhancing service availability of SaaS providers' systems.

KEYWORDS: Anomaly Detection, Performance Prediction, Root Cause Analysis, Impact Analysis.

I. INTRODUCTION

In today's data centers, with the emergence of recent virtualization technologies [8], existing systems are being integrated into a small number of large data centers. A small number of hardware devices are provided to users through private networks or the Internet in such huge data centres, a huge number of applications. This type of facility is called as Software as a Service (SaaS) Platform. It becomes more important than ever before to monitor applications on virtual devices given in this paper, a system that provides this type of service is referred [4].

In SaaS Model, if performance of application affected by physical or virtual resources then it directly affects on other applications which sharing same resources. Therefore, to avoid system failure due to mutual application and resource dependency, application service managers of system have to spend lot of time monitoring the applications related to physical or virtual resources.

In conventional monitoring system regular time series data and event log data are monitored by different monitoring software. Therefore application service managers have to spent time for moving from one monitoring tool to another for collecting and processing data. Regular time series data contains CPU usage, Disk usage, number of active sessions on web servers and they are periodically measured. While event log data contains alerts, starts and ends of batch processing, access of web applications, logins to web portals. Combination of regular time series data and event log data are called as Historical data.

The target of this study states that Integrated monitoring software is designed for collecting and processing of historical data to shorten the transition time (the time to switch from one historical data to another) with five basic data formats.

II. RELATED WORK

Software as a Service is a monitoring service for applications and resources. It supports custom metrics, which include numerical data, statistical values, and string key-value pairs called dimensions. These services and tools have high flexibility for representing structured historical data. However, historical data processing for reducing transition times are out of the scope of the services and tools. They are useful for implementing the integrated monitoring software. In these system use sales and purchase module data set. It gives users related data with respect to their transaction of purchase items.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

This Platform is used for design of integrated system. Here it classifies log level monitoring like error log, system log etc. The classification of log is based on type of historical data. It includes all logs which affect on system performance. It gives access logs of users, authenticate user's logs, number of active users log, access count of web application, transaction logs and root cause of failure logs. The extraction of log is based on some data mining techniques (i.e. Naïve Bayes classification algorithm.)

PROPOSED INTEGRATED MONITORING SYSTEM

The application service managers perform various different kinds of tasks called monitoring tasks. There are majorly five different types of monitoring tasks such as Failure detection, Anomaly detection, Root cause analysis, Performance prediction and Impact analysis. These five major monitoring tasks have been defined [1]. Figure 1 shows working of the monitoring tasks.

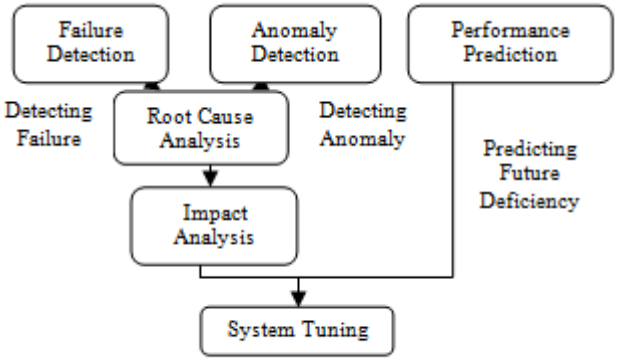

Figure 1.Working of Monitoring Tasks.

A. Failure Detection

Failures cause serious problems on system's service like service shutdown or degradation of quality of service. Failures are checked manually by operators or received an alert from monitoring software. If a failure is detected, they inform infrastructure or service managers about the failure. If any application server or database server is not running (i.e. unplanned downtime).Then it is critical problem. It directly affects service quality. Then in these case alert generated by monitoring software so as to take correct action with respect to failure immediately.

B. Anomaly Detection

Anomaly does not cause serious problems. Anomaly has been defined as variation from the normal or usual order. The normal behavior of system performance indicator values is less than threshold value. If performance indicator values reach to threshold and exceeded from normal value, then anomaly is detected. Here it checks any unauthenticated users tried to access transaction activity. If any unauthenticated user is detected, then system will be blocked that user.

C. Performance Prediction

The system performance degradation and then plan to compare and change system configuration or buy additional hardware with expected growth of load in near future, the application service managers predict performance deficiency. For example if CPU utilization is 95% in today's date. Then after some time or tomorrow there is capability that it may be goes up to 100 %. Due to these downtime of system will occur. The Solution to resolved performance problem is increase the additional ram or use additional processor. The Monitoring software predicts the performance of system with the performance indicator's threshold value; check the increasing tendency of system load in recent month and with the overall system log analysis. It also predicts system performance as considering response time as parameter. It checks delay in response time with respect to transaction performed by user.

D. Root Cause Analysis

Application service managers analyze actual the cause of occurrence of anomaly or failure. In order to study main cause, most of times, managers have to search minute details such as old logs analysis and process level statistics of historical data It also involves run-time analysis, code profiling and need real-time process level statistics to monitor. If any transaction activity is failed, then system will check cause of failures like long running transaction, unauthenticated user, incorrect security pin, system downtime and so on.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

E. Impact Analysis

From the point of view of hardware, software and customers data, application service managers analyze the impact of occurrence of failure or anomaly on system. For example, if any transaction is fail, then the managers have to find services and customers affected by the failure.

III. OPERATIONS PERFORMING ON HISTORICAL DATA

There are mainly two types of operation performing on historical data by application service managers and they are collection of historical data and processing of historical data. They are described are as follows.

A. Collection of Historical Data

The Examples of historical data about application services are listed in Table I. The application service managers monitor historical data about hardware and popular software by specific monitoring tools.

TABLE I. EXAMPLES OF HISTORICAL DATA

Types of historical data		Examples
Data about hardware and OS		CPU usage, memory usage, network bandwidth usage, disk usage.
Data about middleware		Performance data such as number of active connections to database, available memory and number of active sessions.
Data about applications	Batch processing	Start time, execution time, number of processed data and user uploaded data of each batch process.
	Web application	Access count and execution time of each web page or function, detailed logs and login histories of each web application.

B. Processing of Historical Data

To perform the monitoring tasks, processing of historical is essential for application service managers. Processing of historical data is further divided into two types and they are described as follows:

1) Well-Organized Historical Data

In these processing for creating well organized histrical data,there is need for developing program when each new service is released. Therefore application service managers therefore spend a long time performing their monitoring tasks for the new service until these programs are prepared. For example, line charts, indexes for filtering and sorting logs, and statistical values from regular time-series data and event log data are needed.

2) Relation Data for Historical Data

In these processing applications service managers insert characters that represent functionality and quality of services of each host into their hostnames. They also customize an alignment sequence of historical data on conventional monitoring software to easily associate the relations between the data. For example, relations between services and physical/ virtual servers, relations between load balancers and web servers, relations between web servers and database servers and connections between network ports of physical/virtual devices are needed.

The time required for collecting and processing of historical data for performing monitoring tasks are called as “operating time”. Here there are two periodic monitoring tasks and they are anomaly detection and performance prediction. Root cause analysis and system tuning are irregular monitoring tasks. Periodic and irregular monitoring tasks are

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

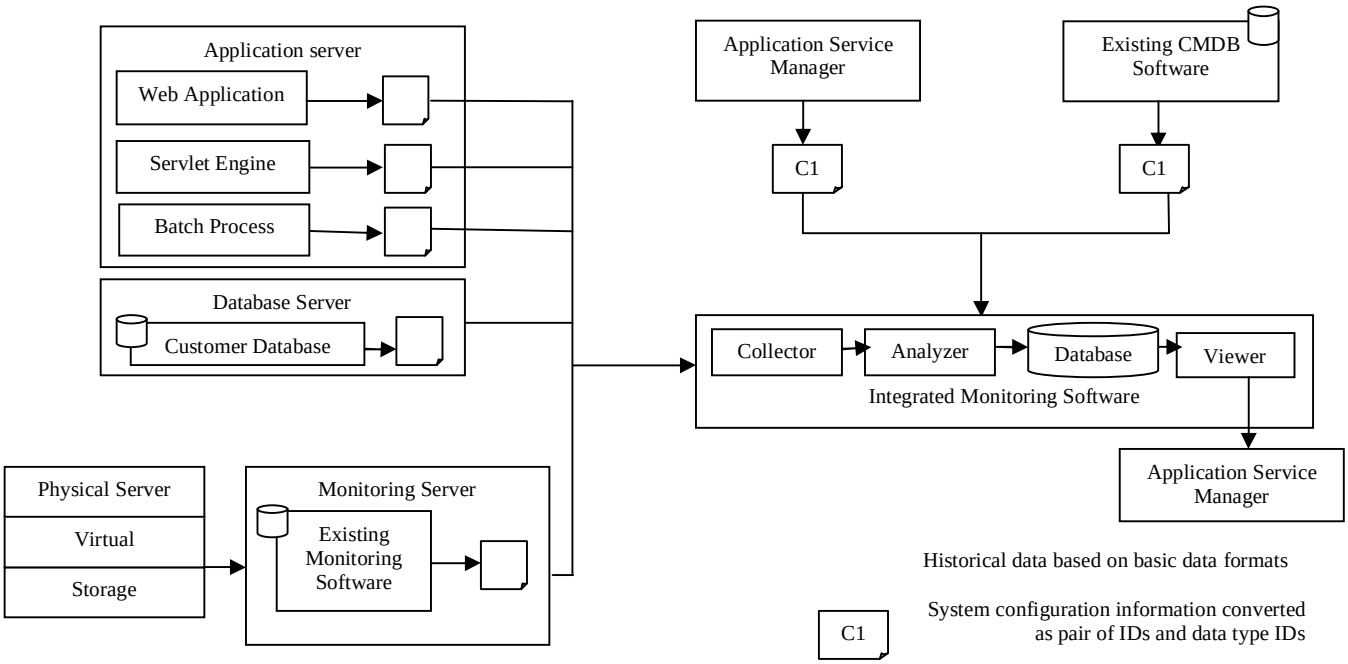

Figure 1. System Architecture of Monitoring Software for Application Service Manager.

time consuming.Therefore, It is difficult for application service managers to find indication of system failures and deterioration in performance rapidly. These adversly affect on shortening transition time.

IV. SYSTEM ARCHITECTURE

System architecture of monitoring software for application service managers is shown in Figure 2. Integrated monitoring software consists of collector, analyzer, database and viewer. Collection of historical with five basic data formats are done by collector. Creation of well-organized historical data and relation data are performed by analyzer. Then these data are inserted them into database. These data are then displayed to application service manager by viewer. Servlet Engine outputs historical data about application and middleware according to basic data formats. Database server converts customer data into basic data formats. Monitoring server converts existing physical or virtual services of data into basic data formats. All these data about applications and physical or virtual services are converted into basic data formats and collected into integrated monitoring software. Application service managers converted configuration information documents into pairs of IDs. Management server converted existing configuration information into the pairs of IDs with existing management software such as CMDB (i.e., Configuration Management Database software).

A. Methods for Displaying Historical Data

To display historical data, the integrated monitoring software needs to be able to read all major historical data. The integrated monitoring software needs to provide similar display functions as those of existing monitoring software and

transition functions among the historical data using system configuration data.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

B. Basic Data Formats of Historical Data

For monitoring application service managers, there is need for historical data about a wide range of applications and physical/virtual resources. There are defined historical data which contains minimum necessary data for five basic data formats. The five basic data formats are Performance data, Response data, Command execution data, Web event data and Login history. Processing of only performance data and response data are supported by conventional monitoring software. In addition to these basic data formats integrated monitoring software supports three basic data formats and they are command execution data, web event log and login history. There are three mandatory fields for all basic data formats. They are necessary for analyzer to search relations between historical data using system configurations .They are using two types of identifier as mandatory fields i.e. data type ID and Instance ID for monitoring physical/virtual resources and applications. Third mandatory field is timestamp which shows time in which historical data is created.

The performance data is a data type for representing a simple performance value. The data collected by existing monitoring software are converted into performance data. The command execution data is a data type for representing the detailed result of batch processing. The web event log is a data type for representing customers' operations concerning web applications and execution times measured at the web servers. It can include a number of descriptions about the customer's operations. The login history is a data type for representing the login time and the logout time of each customer. The response data is a data type for representing response times measured by multiple remote sites.

As shown in Figure 2 analyzer performs background processing of historical data. It includes both creation of well format historical data and their relational data for historical data. After collecting data from collector analyzer arranges these data into well-organized format and also manages relations between historical data like relations between pairs of instance ID and data type ID. System Tuning or configuration information's are necessary for creation of relation data. Application service managers created this information manually or converted data from existing management software (CMDB) such as pairs of instance IDs, pairs of data type IDs, or combination of both.

V. RESULTS AND EVALUATIONS

The results are observed using survey held by application service managers [6].

A. Evaluations

There are 3 types of evaluation methods are used and they are briefly discussed below.

1) Who performs Monitoring Tasks

All monitoring tasks are performed by application service managers in order to check system performance.

2) Questions in the survey

In these questionnaire survey, application service managers asked questions about how many monitoring tasks are involved and what they do, users service access intensity when failure or anomaly occurs, finding timestamp of batch processing execution times, current use of monitoring software and what action to be taken when anomaly is detected and so on.

3) Transition Times Estimation

The third evaluation method is to estimate transition time based on log search time and standard transition time.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

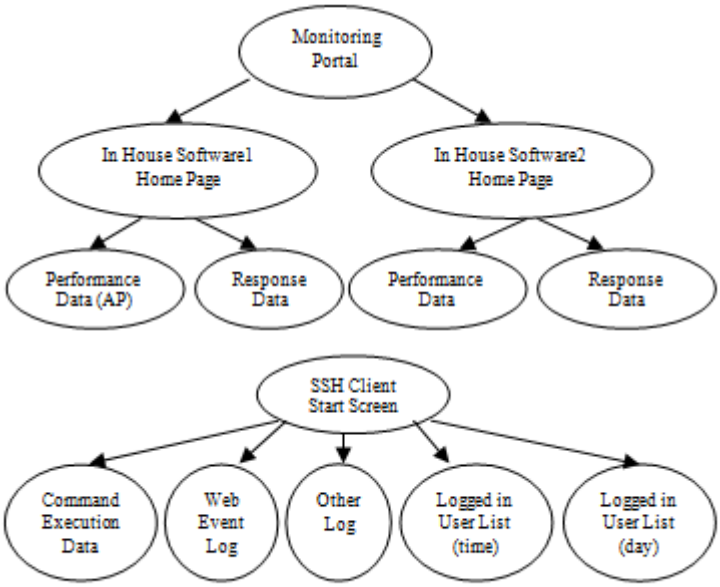

Figure 3.Transition times of Existing System.

It compares existing system transition times with proposed integrated system. In Figure 3 shows transition times of existing system. It includes on performance data and response data of monitoring tasks. So it covers only 69.6 % of monitoring tasks.

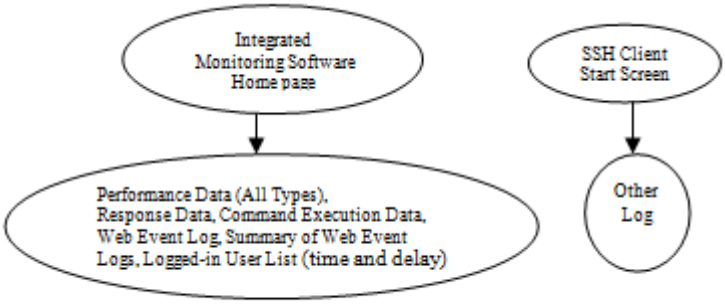

Figure 4.Transition times of Integrated System.

In Figure 4 shows transition times of integrated system. It shows all performance data, response data, command execution data, web event log data, login histories data collectively. So it takes shorten transition time as compared to existing system.

TABLE II. DATA SAMPLE LOGS OF TRANSACTION

Data Sample log	Frequency count	Transition time(in millisecond)	Total transition time for transaction log(T)
Failure due to incorrect Otp(A)	3	210545	1202133
Successful transaction(B)	11	533184	1202133
Failure due to	2	167030	1202133

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

low balance(C)			
Failure due to long processing time(D)	7	290024	1202133
Failure due to invalid login(E)	3	1350	1202133

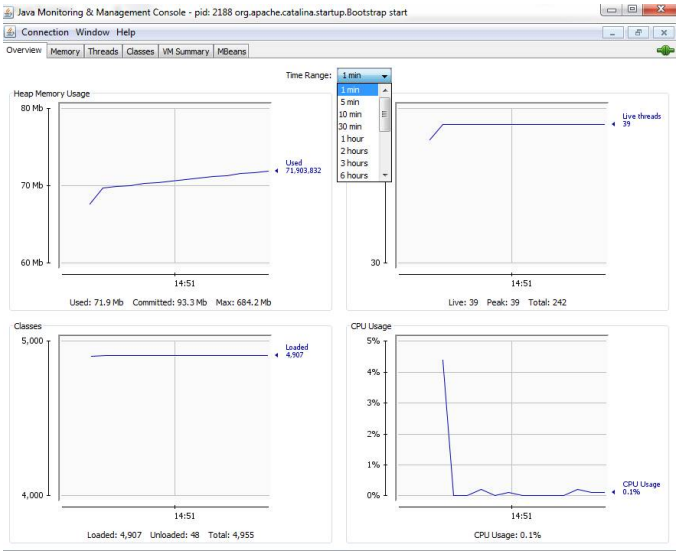

Figure 5.Live result of Resource Monitoring System

TABLE III. PERFORMANCE PREDICTION EVALUATION

Basic data format	Transition on time (Tn)	Transition time Tn=Tn-T(where T=210545)	Performance prediction ((T1-Tn)) /T1*100
A&B	210545	T1=1202133(first)	82.48
A&B&C	3775475	T2=167030	86.10
A&B&D	500569	T3=290024	75.87
A&B&E	211895	T4=1350	99.88
A&B&C&D	667599	T5=457054	61.97
A&B&C&D&E	1350	T6=1350(Last)	99.88

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

TABLE IV. MONITORING TASK COMPLETED IN INTEGRATED SYSTEM

Basic data	Monitoring Tasks					
	Anomaly Detection	Performance Prediction	Root Cause Analysis	Impact Analysis	Other	All Types
A&B	100	82.48	53.84	100	46.15	76.49
A&B&C	100	86.10	61.53	100	38.46	77.22
A&B&D	100	75.87	80.76	100	19.23	75.17
A&B&E	100	99.87	65.38	100	34.61	79.97
A&B&C&D	100	61.97	88.46	100	88.46	87.78
A&B&C&D&E	100	99.88	100	100	91.17	98.16

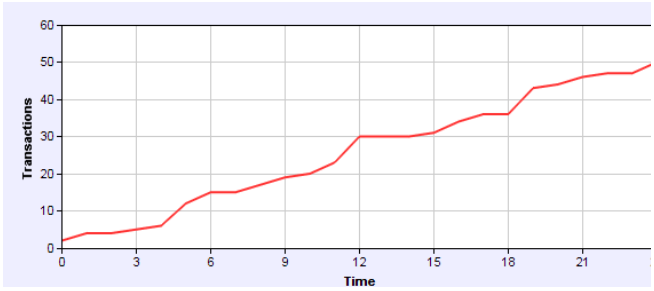

Figure 6.Time required for transactions of Integrated Monitoring software.

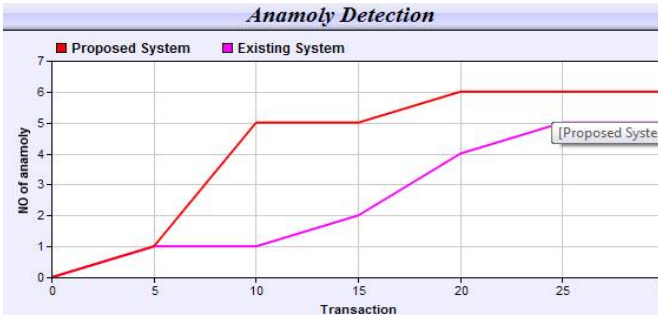

Figure 7.Results of Anomaly Detection in Integrated Monitoring Software.

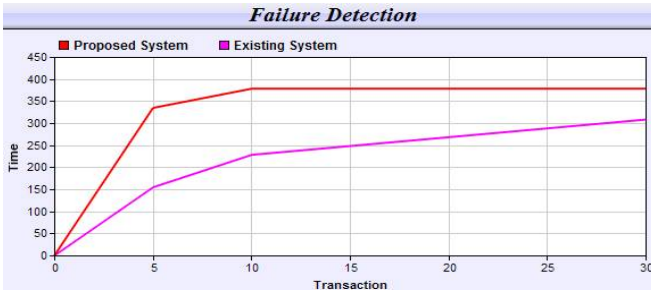

Figure 8.Results of Failure Detection in Integrated Monitoring Software.

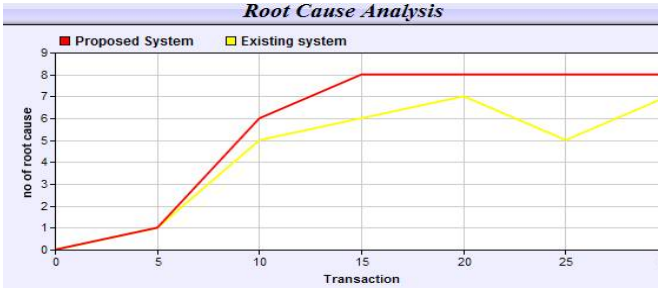

Figure 9.Results of Root Cause Analysis in Integrated Monitoring Software.

B. Results

The conventional monitoring software with basic data formats such as performance data and response data supports about 69.72% of monitoring tasks. In integrated monitoring system analyze live result of resource monitoring .It gives details about how much CPU usage, memory usage, heap memory usage live threads with respect to user specified time in system. Figure 6 shows the graph according to performance time of system. It shows performance of the system with respect to transactions in accordance with time. TABLE II shows percentage of monitoring task completed in existing system with basic data format. A, B, C, D and E indicated basic data format as performance data, response data, command execution data, web log event data, login histories data and other log data respectively. TABLE II gives details about data samples used for basic data format for doing transaction with their frequency count and transition time. Frequency count is calculated by using queries fetched by user from database. TABLE III calculates percentage of performance prediction based on transition time. TABLE IV estimates completeness rate of monitoring tasks completed with basic data format. Root cause analysis is percentage ratio of frequency count of basic data format to total frequency count of transaction log occurred. For example Root cause analysis of A and B is calculated using total frequency count of A and B to the total frequency count of transactional log. Anomaly detection is calculated using

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

total number of anomaly detected in system accurately or not. For example unauthenticated user tried to access system, then system blocked that user for transaction. Anomaly is calculated using number of failure of transaction with respect to that user with specific time. Impact analysis is estimated based on all the transaction activity is detected by system accurately. If any transaction in system affects system continuity then system takes further action so as they didn't affect on system performance. Other log calculated remaining logs of transaction to the total transaction logs of system. In all types calculated average of all monitoring tasks with their basic data format. Integrated monitoring system supports 98.16% of monitoring tasks with all basic data format. It also reduces transition times required for monitoring task compared to existing software. Figure 7 shows number of anomaly detected in transaction with time in both Existing and proposed system. Figure 8 shows comparison between failure detection in existing and proposed system. It shows number of failure detected in system with respect to transaction and time. Figure 9 shows root cause of failure or anomaly detection in existing and proposed integrated system. It detects any anomaly or failure in system with respect transaction. If any anomaly or failure detects, then it tries to find out root cause of failure.

VI. CONCLUSION

Integrated monitoring software evaluated historical data was proposed for shortening operation times. Well-organized historical data before the application service managers' requests could shorten the transition times which surveys on current monitoring tasks for over many companies showed; creating relation data for recommendations could shorten the transition times; and a grouping of the two types of processing had a stronger effect on the transition times than either of them individually. In monitoring software for application service managers recommended to implement both types of processing. The results of survey show that proposed monitoring software with all of the basic data formats supports 98.16% of monitoring tasks. The proposed integrated monitoring software results suggest particularly effective in mission-critical application services.

REFERENCES

[1] Masahiro Yoshizawa, Tatsuya Sato and Ken Naono, "Integrated Monitoring Software for Application Service Managers", IEEE Trans. on Network and Service Management, Vol. 11, No. 3, , pp. 321-332, September 2014.
[2] Alina Girbea, Constantin Suciu, "Design and Implementation of a Service-Oriented Architecture for the Optimization of Industrial Applications", IEEE Trans. on Industrial Informatics, Vol.10, No.1, pp.185-195, February 2014.
[3] Mehdi Mirakhorli, Jane Cleland Huangy, "Detecting, Tracing, and Monitoring Architectural Tactics in Code", IEEE Trans. on Software Engineering, pp. 1-18, 2015.
[4] M. Armbrust, A. Fox, R. Griffith, "Above the clouds: A berkeley view of cloud computing", University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009.[online].Available:
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html.
[5] B. Krishnamurthy, A. Neogi, "Data tagging architecture for system monitoring in dynamic environments", In Proc. NOMS, 2, pp. 395–402, 2008.
[6] M. Yoshizawa and K. Naono, "Design and evaluation of integrated monitoring software for SaaS-based systems", In Proc. APNOMS, pp. 1–4, 2011.
[7] Xu Han, Suvasri Mandal, "An Optimization-Based Distributed Planning Algorithm: A Blackboard-Based Collaborative Framework", IEEE Trans. on systems, VOL. 44, NO. 6, pp.673-686, JUNE 2014.
[8] VMware vSphere. [Online]. Available: http://www.vmware.com/products/vsphere/
[9] Eser Kandogan, Aruna Balakrishnan, "From Data to Insight Work Practices of Analysts in the Enterprise", IEEE Computer Society, pp. 42-50. October 2014.
[10] Tapan Nayak, Anindya Neogi, " Visualization and Analysis of System Monitoring Data using Multi-Resolution Context Information", IEEE Trans. On Network and Service Management, VOL. 5, No. 3, pp. 162 -177, September 2008.
[11] P. Kavita, M. Usha, " Anomaly Based Intrustion Detection" , Jounal of theoritical and applied IT, Vol.61,No. 3, March 2014.