IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 10, October 2017

Real Time Monitoring and Data Analytics of IoT Data Servers

Rohit Tapas[1], Prasoon Maurya[1], Dr. Poorna Shankar[2]

B.E. Department of Computer Engineering, Indira College of Engineering and Management, Pune, India[1]

HOD, Department of Computer Engineering, Indira College of Engineering and Management, Pune, India[2]

Abstract: In businesses today it is very important to understand the importance of real time monitoring of the data servers that are connected through internet and distributed globally. Such servers holds the real-time enterprise data which will be used to improve the business. To ensure peak performance of applications running on servers, the server hardware must be working well, the servers should be sized well to handle their workload, and there should be no resource bottlenecks. Hence, we have to make sure that these servers work properly without any faults. The servers should be properly monitored to ensure optimum throughput from the system. Server Monitoring helps understanding servers' system resource usage which can help in capacity planning and provide a better end-user experience. Logging can also be monitored to analyze the event occurrences. Performance and configuration monitoring is also mandatory to reduce the downtime of servers and increase the efficiency of processors. Server monitoring provides the data relating to operating system and when used in conjunction with other monitoring data from the application a true glimpse can be obtained into the working of the system. Our proposed system monitors services like CPU Usage, Memory Consumption, I/O, Network, Disk Usage, Processes, Component reachability etc. Moreover, alerts can be sent to authorities in various forms when a specified event takes place. Thus, the concerned authorities can make changes to keep the system up and running.

Keywords : Server, Monitoring, Services, Resources, Faults, Alerts

I. INTRODUCTION

A server is a computer designed to process requests and deliver data to other (client) computers over a local network or the Internet In recent times almost all companies regardless of their size, use servers. In fact servers have become an integral part for the functioning of an organization. Servers of an organization are crucial for its day to day work. In this paper all distributed IOT commercial servers are monitored through Nagios which is a open source network and server monitoring software. It is popular because of the flexibility to monitor the servers with both agent-based and agentless monitoring. It quickly detects and identifies the problems that are pertaining to servers, network and processes in windows server, linux server and Unix server.

A. Server monitoring:

Server Monitoring is a process to monitor server's system resources like CPU Usage, Memory Consumption, I/O, Network, Disk Usage, Process etc. Server Monitoring helps in understanding server's system resource usage and supports in capacity planning and provide a better end-user experience. The structure of Nagios is depicted below.

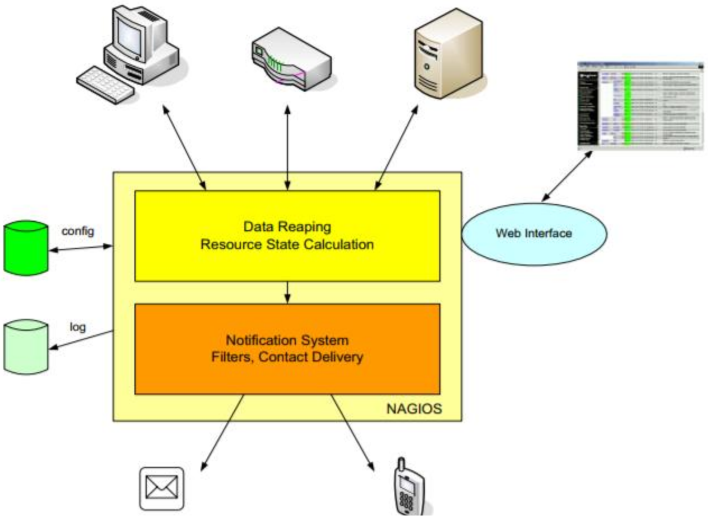


Fig. 1. Nagios Structure

DOI 10.17148/IJARCCE.2017.61064

Managing the increasing number of servers with the limited resources is a big challenge for server monitoring. As soon as one server cluster gets up and running, there is need to setup another one. But with the limited number of resources in the organization, server management and monitoring becomes difficult. IT infrastructure with growing number of servers also becomes heterogeneous. Different servers have different operating systems and thus need different methods for implementation, management and monitoring. It is also important that the Server Monitoring gains deep service level insights into the systems for a complete coverage and monitoring. Server Monitoring should be able to ensure that the system is functionally stable and working efficiently. Due to the load on the servers, it is often overlooked that the hardware resources might be running out. Server monitoring should be able to help in effective management of resource utilization by giving insights into resources.

B. Significance of Server Monitoring:

Real time monitoring not only gives us accurate on the fly information of how good or bad our network environment is running but also hugely helps with foreseeing any future possible issues which the network might face as well as troubleshoot any on-going support work required. Effective real time monitoring not only is crucial to most business because of the importance of critical application monitoring ensuring it is functioning properly but also saves businesses money in the long run if any application/server downtime were to occur as it bringing up the server environment wouldn't be difficult knowing there is full traceability as to what happened before the issue occurred.

C. Monitoring through Nagios:

Nagios is an open source monitoring system for computer systems. It was designed with core components to run on the Linux operating system and can monitor devices running Linux, Windows and Unix OSes. Nagios runs periodic checks on critical parameters of application, network and server resources. It can monitor, for example, memory usage, disk usage, microprocessor load, the number of currently running processes and log files. Nagios also can monitor services, such as Simple Mail Transfer Protocol (SMTP), Post Office Protocol 3 (POP3), Hypertext Transfer Protocol (HTTP) and other common network protocols. Active checks are initiated by Nagios, while passive checks come from external applications connected to the monitoring tool.

D. Nagios structure

Nagios is organized as a pluggable, open source tool, which makes it easy to develop new components for it and to extend its functionality. At the heart of Nagiosis its server, where plug-ins and add-ons allow the user to define targets and which parameters on these targets to monitor. For example, when used in conjunction with environmental-sensing systems, Nagios can share data on environmental variables, such as temperature, humidity or barometric pressure.Nagios can also run remote scripts by using the Nagios Remote Plugin Executor, also called NRPE.Nagios runs in agent-based and agentless configurations. The user can install a Nagios monitoring agent on any resource they wish to track, or rely on agentless monitoring protocols to track performance. The choice between agent-based and agentless monitoringdepends on the design of the IT infrastructure and desired monitoring setup.

E. Services Monitoring Through Monit :

The monit utility is a simple lightweight monitoring Open Source tool for managing and monitoring processes, programs, files, directories and filesystems on a Unix system. Monit is controlled via an easy to configure control file based on a free-format, token-oriented syntax. Monit logs to syslog or to its own log file and notifies about error conditions via customizable alert messages. Monit utility can also be used to send email alerts through SMTP server . Monit has a ability to start a process if it is not running, restart a process if not responding and stop a process if uses high resources. Additionally you can also use Monit to Monitor files, directories and filesystems for changes, checksum changes, file size changes or timestamp changes. With Monit you can able to monitor remote hosts TCP/IP port, server protocols and ping. Monit keeps its own log file and alerts about any critical error conditions and recovery status.

II. LITERATURE SURVEY

Due to the complexity of nowadays networks and the inadequate of existing open-source software features,network administrators usually have to integrate several tools to build up the monitoring environments that meet their requirements. Nagios is one of those tools that have been widely used by experienced network administrators. Because of the flexible modular architecture,Nagios allows users to develop custom modules to enhance the system functionality in many different ways. In this paper, we propose the conceptual design of the seamless integration of Nagios as a core of the new feature-rich monitoring system. Our new system is integrated with a more interactive and friendly user interface, while providing much more in-depth information about the network. Most importantly, all of which can be achieved without modifying any single line of Nagios source code.[1]

DOI 10.17148/IJARCCE.2017.61064

A new feature of services in Nagios has been added to the existing system which has no such services. The bandwidth monitoring and notification system are configured for alerting the network administrators when the bandwidth of the network in an organization hits a certain threshold settings. The system sent an email alert and sms notification to the network administrator for taking further action in order to maintain the Quality of Service (QoS) in the network. All the logs file of the Nagios actions is saved in the NagiosFile Logs. The analysis was conducted from the case study and problem statements. Network Development Life Cycle (NDLC) was chosen as a methodology for implementing this system in the network. Nagios is installed inside Ubuntu 10 Operating System along with Multi-Router Traffic Grapher (MRTG) and Mail Postfix. MRTG and Mail Postfix were configured to be integrated with the Nagios System. On the client side, NSClient++ has been installed, for monitoring the bandwidth and performance of windows based on operating system. The Nagios services have been improved with the implementation of sms and emails notifications since the existing services have no such utilities. With the implementation of these services to Nagios, the performance could be even better for the future. [2]

Over the past years services computing has become an emerging science that is highly regarded as a necessary technology not only by research but by industry as well. In the same context, the advent of cloud computing gave to services and web applications a whole new perspective and potential. Regardless of the rapid evolution in the fields of services and web technologies, ensuring the QoS of computing resources still remains an important topic. To this end, monitoring computing resources and application execution is an integral part of the services computing value chain. In this paper we present the architectural design and implementation of a service framework that monitors the resources of a physical as well as virtual infrastructure. Our solution extends Nagios, a widely used monitoring toolkit, through the implementation of NEB2REST, a Restful Event Brokering module. [3]

A design for a new dynamically reconfigurable distributed modular monitoring system framework is proposed in this paper. The proposed design allows combining both monitoring tasks (supercomputer ‘health’ monitoring and performance monitoring) in one monitoring system. Our approach allows different parts of the monitoring system process only the data needed for the task assigned to these parts. This helps to process a lot of performance data and to get information about dynamic features of heavy parallel tasks. Another feature of our framework is the ability to calculate performance metrics on-the-fly, dynamically creating processing modules for every job or other objects of interest.[4]

This paper is focused on evaluation of Nagios, an open-source flexible monitoring tool for enterprises, in a cloud-based network orchestrated by OpenStack. The work relies on a previous work done within UC Labs which added new objects (i.e. QoS parameters) in a management information base MIB. The solution aims at collecting the Available Transfer Rates and One-Way Delays of the links within the cloud, in order to provide the optimization of the Network Virtualization Functions. [5]

This system is able to detect and report failures of devices, services or connections and send messages (alerts) to designated locations to notify system administrators so as to ensure the proper functionality of the services. All events that changing the operational status of the network are recorded and stored in a database. Implementing a proper mathematical formula is able to calculate the availability of the separate SDN-II services as well as the total availability index of the network. The calculation this index is a mandatory key-point for any service provider (including SDN-II) to be able to signService Level Agreenets (SLAs). [6]

The paper is focused on Nagios, an open-source flexible monitoring tool for enterprises. The goal was to add new objects (i.e. QoS parameters) in a management information base MIB starting with the root object identifier. Furthermore, we wanted to prove that the integration of this enhanced software tool running on several platforms (Android, Windows7, Fedora Core) is a benefit for network administrators. The newly created SNMP Agent and with the existing Nagios Agent were able to update the parameters: Available Transfer Rate ATR and One-Way-Delay OWD. Due to cross-layer techniques involved, the overall solution is an evolutionary step forward towards Future Internet implementations (although it is still stick on SNMP approach). [7]

III. PROPOSED SYSTEM

Our proposed system has the following features:

A. Failure of Service:

When a fault occurs, services may stop working. This results in failure of system. Thus we monitor the servers on which the services are running and prevent the services from failing.

DOI 10.17148/IJARCCE.2017.61064

B. Automatic Restarting:
If any service goes down, it will restart without any human intervention.

C. Alert Generation:
When a service fails or a specified threshold value is reached, alerts are sent to the authorized personnel via SMS and e-mail.

D. Real Time Graphs:
Real time graphs depicting statistics of the servers will be displayed to the user.

E. Creation of Interface:
An interface which can remotely control all the monitored services on the servers from one place.

IV. ARCHITECTURE

There are many organizations in the world. Gigabytes of data is transferred using the internet. Such orgs require tools to monitor the data stored on their servers. Tools like Monit, Nagios help in monitoring these servers and provide alerts using sms or email. After completion of monitoring the servers, our next task is to generate graphs so as to provide our clients a brief idea about when their server might get shut down. This can be accomplished using LOGS and various data mining algorithms

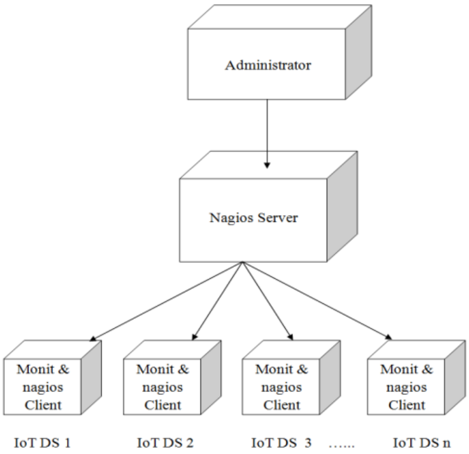
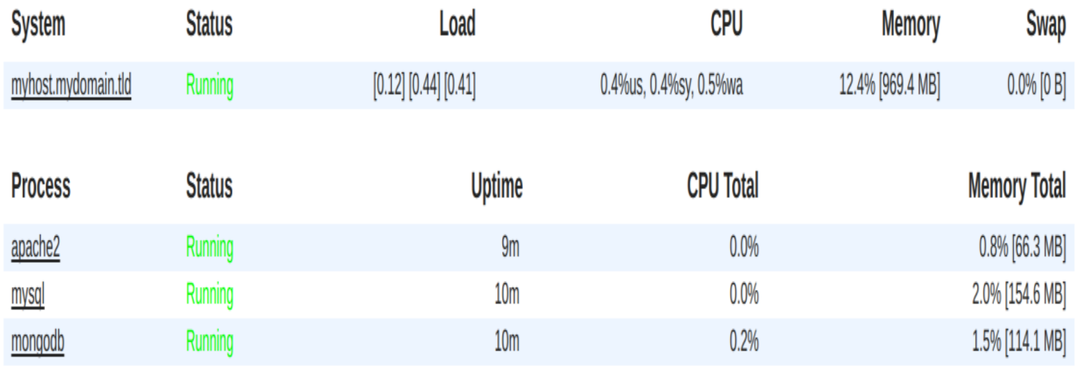


Fig 2. Deployment Architecture

A. Advantages:

1. Automatic detection of any kind of server failures
2. No need of manual restart of the server
3. Automated alert generation in real time
4. Real Time Data Analysis
5. Easy to Use
6. 24-7 updated status results of system

Screenshots:



System	Status	Load	CPU	Memory	Swap
myhost.mydomain.tld	Running	[0.12] [0.44] [0.41]	0.4%us, 0.4%sy, 0.5%wa	12.4% [969.4 MB]	0.0% [0 B]

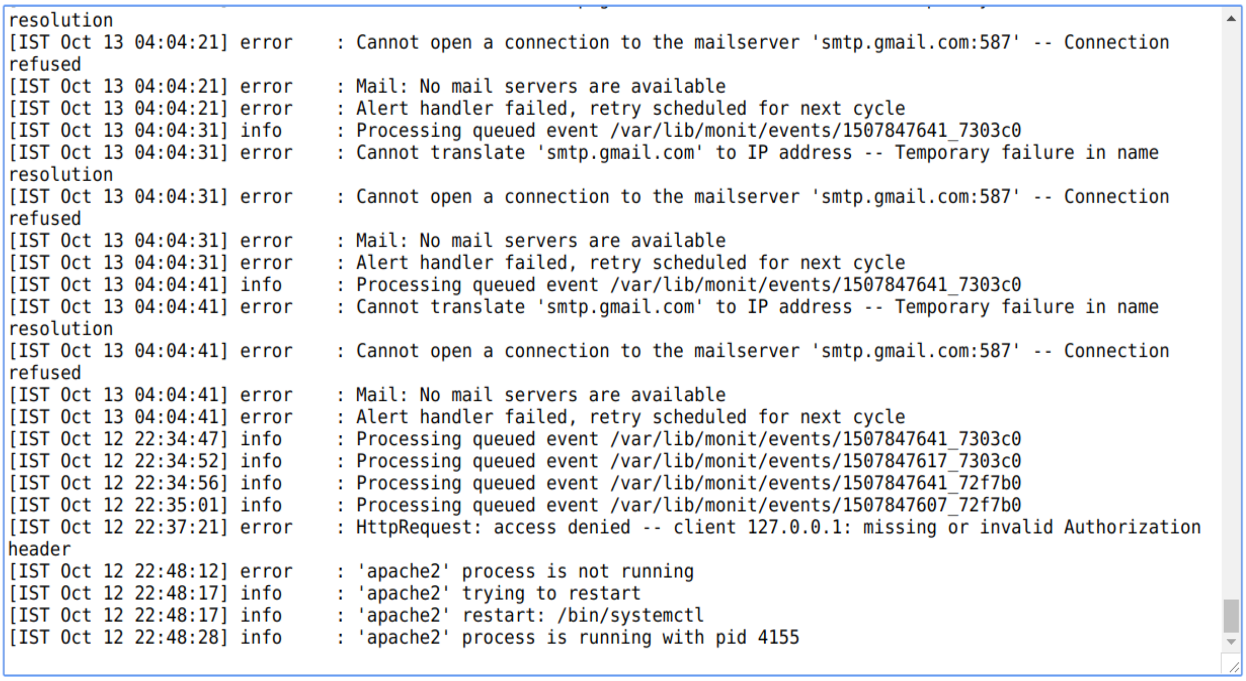
Process	Status	Uptime	CPU Total	Memory Total
apache2	Running	9m	0.0%	0.8% [66.3 MB]
mysql	Running	10m	0.0%	2.0% [154.6 MB]
mongodb	Running	10m	0.2%	1.5% [114.1 MB]

Fig. 3 Monitoring Status

DOI 10.17148/IJARCCE.2017.61064

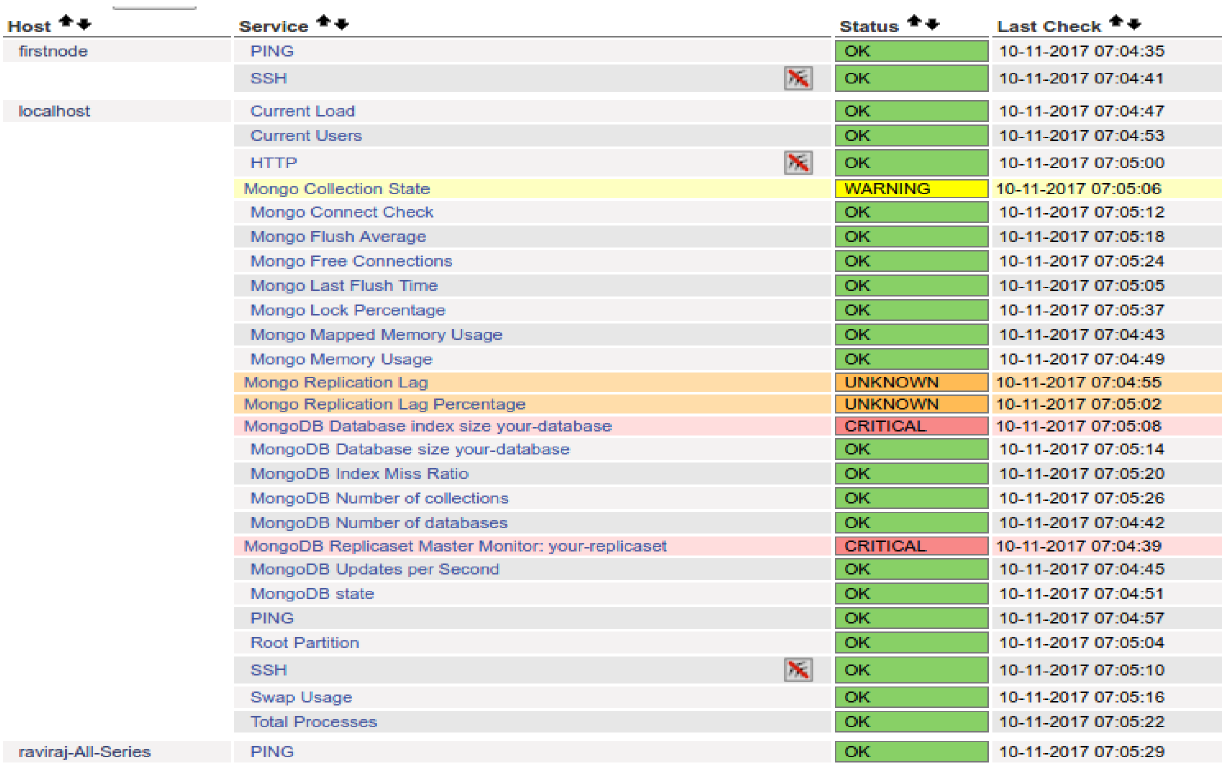
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 10, October 2017



```
resolution
[IST Oct 13 04:04:21] error    : Cannot open a connection to the mailserver 'smtp.gmail.com:587' -- Connection
refused
[IST Oct 13 04:04:21] error    : Mail: No mail servers are available
[IST Oct 13 04:04:21] error    : Alert handler failed, retry scheduled for next cycle
[IST Oct 13 04:04:31] info     : Processing queued event /var/lib/monit/events/1507847641_7303c0
[IST Oct 13 04:04:31] error    : Cannot translate 'smtp.gmail.com' to IP address -- Temporary failure in name
resolution
[IST Oct 13 04:04:31] error    : Cannot open a connection to the mailserver 'smtp.gmail.com:587' -- Connection
refused
[IST Oct 13 04:04:31] error    : Mail: No mail servers are available
[IST Oct 13 04:04:31] error    : Alert handler failed, retry scheduled for next cycle
[IST Oct 13 04:04:41] info     : Processing queued event /var/lib/monit/events/1507847641_7303c0
[IST Oct 13 04:04:41] error    : Cannot translate 'smtp.gmail.com' to IP address -- Temporary failure in name
resolution
[IST Oct 13 04:04:41] error    : Cannot open a connection to the mailserver 'smtp.gmail.com:587' -- Connection
refused
[IST Oct 13 04:04:41] error    : Mail: No mail servers are available
[IST Oct 13 04:04:41] error    : Alert handler failed, retry scheduled for next cycle
[IST Oct 12 22:34:47] info     : Processing queued event /var/lib/monit/events/1507847641_7303c0
[IST Oct 12 22:34:52] info     : Processing queued event /var/lib/monit/events/1507847617_7303c0
[IST Oct 12 22:34:56] info     : Processing queued event /var/lib/monit/events/1507847641_72f7b0
[IST Oct 12 22:35:01] info     : Processing queued event /var/lib/monit/events/1507847607_72f7b0
[IST Oct 12 22:37:21] error    : HttpRequest: access denied -- client 127.0.0.1: missing or invalid Authorization
header
[IST Oct 12 22:48:12] error    : 'apache2' process is not running
[IST Oct 12 22:48:17] info     : 'apache2' trying to restart
[IST Oct 12 22:48:17] info     : 'apache2' restart: /bin/systemctl
[IST Oct 12 22:48:28] info     : 'apache2' process is running with pid 4155
```

Fig. 4 Error and Information Logs



Host	Service	Status	Last Check
firstnode	PING	OK	10-11-2017 07:04:35
	SSH ☒	OK	10-11-2017 07:04:41
localhost	Current Load	OK	10-11-2017 07:04:47
	Current Users	OK	10-11-2017 07:04:53
	HTTP ☒	OK	10-11-2017 07:05:00
	Mongo Collection State	WARNING	10-11-2017 07:05:06
	Mongo Connect Check	OK	10-11-2017 07:05:12
	Mongo Flush Average	OK	10-11-2017 07:05:18
	Mongo Free Connections	OK	10-11-2017 07:05:24
	Mongo Last Flush Time	OK	10-11-2017 07:05:05
	Mongo Lock Percentage	OK	10-11-2017 07:05:37
	Mongo Mapped Memory Usage	OK	10-11-2017 07:04:43
	Mongo Memory Usage	OK	10-11-2017 07:04:49
	Mongo Replication Lag	UNKNOWN	10-11-2017 07:04:55
	Mongo Replication Lag Percentage	UNKNOWN	10-11-2017 07:05:02
	MongoDB Database index size your-database	CRITICAL	10-11-2017 07:05:08
	MongoDB Database size your-database	OK	10-11-2017 07:05:14
	MongoDB Index Miss Ratio	OK	10-11-2017 07:05:20
	MongoDB Number of collections	OK	10-11-2017 07:05:26
	MongoDB Number of databases	OK	10-11-2017 07:04:42
	MongoDB Replicaset Master Monitor: your-replicaset	CRITICAL	10-11-2017 07:04:39
	MongoDB Updates per Second	OK	10-11-2017 07:04:45
	MongoDB state	OK	10-11-2017 07:04:51
	PING	OK	10-11-2017 07:04:57
	Root Partition	OK	10-11-2017 07:05:04
	SSH ☒	OK	10-11-2017 07:05:10
	Swap Usage	OK	10-11-2017 07:05:16
	Total Processes	OK	10-11-2017 07:05:22
raviraj-All-Series	PING	OK	10-11-2017 07:05:29

Fig. 5 Monitoring Using Nagios

V. CONCLUSION

Unified monitoring of all server hardware processes, Logged data performance of applications, network and services such as CPU utilization, memory usage and file system are observed from one console and notified to the authorities through alerts. The system also generated graphs of all the monitored data. Remote monitoring of services running on servers in real time is achieved with Monit. Upon failure, services are restarted without human intervention. Alerts and warning messages are generated and sent to the concerned authorities on occurrence of a failure event. Remote monitoring of servers is achieved with Nagios. Nagios allows the user to define targets and which parameters on these targets to monitor. Statistical analysis of log files is done in order to find patterns.

DOI 10.17148/IJARCCE.2017.61064

VI. FUTURE SCOPE

Future work includes alerts using automated VoIP calls to the concerned authority. Users would be equipped to take predefined actions on specified key-press during the call.

Cross platform mobile application will be developed which will enable monitoring with the help of smartphones.

REFERENCES

[1] S.M. Magda, A.B. Rus, V. Dobrota, "Nagios-Based Network Management for Android, Windows and Fedora Core Terminals Using Net-SNMP Agents", 11th RoEduNet International Conference "Networking in Education and Research", Sinaia, Romania, January 17- 19, 2013, ARNIEC/RoEduNet Agency, IEEE Romanian Section, "Politehnica" University of Bucharest, Ministry of Education and Research, ISSN-L 2068-1038, pp. 115-120, DOI: 10.1109/RoEduNet.2013.6511742

[2] V. Dobrota, A.B. Rus, "Cross-Layer QoS Implementation: Clean-Slate Approach", pp.53-93, DOI: 10.4018/978-1-4666-0960-0.ch003, ISBN13: 9781466609600, ISBN10: 1466609605, EISBN13: 9781466609617, in Habib F. Rashvand&Yousef S. Kavian (editors), Using Cross-Layer Techniques for Communication Systems, IGI Global, April 2012, 404 p., DOI: 10.4018/978-1-4666-0960-0, ISBN13: 978-1- 466609600.

[3] F. Rossigneux, L. Lefevre, J.P. Gelas, D. Assuncao, and M. Dias, "A generic and extensible framework for monitoring energy consumption of OpenStack clouds", IEEE Fourth International Conference on Big Data and Cloud Computing BdCloud 2014, pp. 696-702, 2014

[4] http://docs.openstack.org/developer/ceilometer/architecture.html

[5] K. Benz, "A NagiosOpenStack Integration & Data Collection Tool", Zurich University of Applied Sciences, 2015

[6]"Technical Overview | Open Platform for NFV (OPNFV)", Opnfv.org, 2016. [Online]. Available: https://www.opnfv.org/software/technicaloverview. [Accessed: 04- Mar - 2016].

[7]Wireshark,availableon https://www.wireshark.org/

DOI 10.17148/IJARCCE.2017.61064