A Framework for Centralized Access Monitoring over Cloud Architectures

Ajay Prasad
University of Petroleum and Energy Studies
Dehradun, India

Prasun Chakrabarti
Sir Padampat Singhania University
Udaipur, India

ABSTRACT
While we talk about monitoring, the performance and compliance aspects are only on focus, however, the fine grained access logs also needs to be maintained if a proper internal audit is to be realized at organization level. Also, we do need to maintain long term logs for auditing purposes at internal auditing. In generic monitoring services provided by most of the cloud vendors, the case of monitoring users at organization level is not covered so far. It is also to be realized that internally cloud users (as an organization) can delegate the leased resources to its internal employees. In such cases the auditors would require the usage logs having various audit parameters. If the logs maintained at user levels can be verified with the logs maintained at cloud level, then the trust among the cloud vendors and the user will go up by almost hundred percent. The setup of a cloud monitor model along with centralized, verifiable and long term logs can be realized by a cloud framework presented hereby. The paper discusses the viability of the proposed framework over popular open source monitoring tools available and used by many cloud architectures.

Keywords
Cloud computing, centralized access monitoring, monitoring framework, CMaaS.

1. INTRODUCTION
Most of the cloud computing frameworks including openstack [1] and open nebula [2] provides a minimal system of monitoring. However, in all the cases, the monitoring is mostly confined to basic parameters like availability, performance and QoS. The monitoring is not based on end user usage if the organization has many users. The organizational users were mostly monitored very closely in the traditional proprietary network models. Also, with the advent of cloud computing, most of the organizations are switching onto cloud. But organizations as users will have limited control over the usage of respective services and resources by the end employees. It thus, becomes vital to have a framework which extends the generic monitoring interfaces to a more specific and usage based monitoring frameworks. In other words, we are talking about fine grained access monitoring support in cloud computing. The Openstack architecture [1] has a module of Nagios [3] which performs the tasks of monitoring. Similarly, open nebula architecture [2] utilizes Ganglia [4] modules for monitoring. Figure 1 a) and b) depicts the nagios and ganglia architectures respectively.

Nagios is composed of 3 parts [5] i) scheduler ii) a Gui iii) plugins. Nagios functions in an agent based client-server architecture. The client agents are the plugins installed on the monitored hosts. The plugin send information to server which displays them in a GUI. The Nagios monitor server can be integrated or interfaced using the plugin APIs at every host cluster. The Ganglia comprises of GMOND and GMETAD. The Ganglia architecture is also similar to Nagios in the way of client agent-server approach. The Ganglia Monitor daemon (GMOND) [6] is the data collecting agent that is installed at every node to be monitored and a centralized server having Ganglia METAdata daemon (GMETAD) collects the data from various agents and consolidates them.

In both Nagios and Ganglia the logging metrics can be defined by the users. However, the metrics that are sent by the agents can be limited to only the availability, QoS and SLA parameters of the hosts. The resource access from cloud users might not be available to the plugins directly. However, a framework can be constructed which can manage metrics and add access logs metadata to the metrics or add metrics received from the servers to the access log metadata. The Centralized (access) Monitoring as a Service (CMaaS) framework presented herewith can utilize the popular monitoring frameworks to have a fine grained access monitoring over the cloud services.

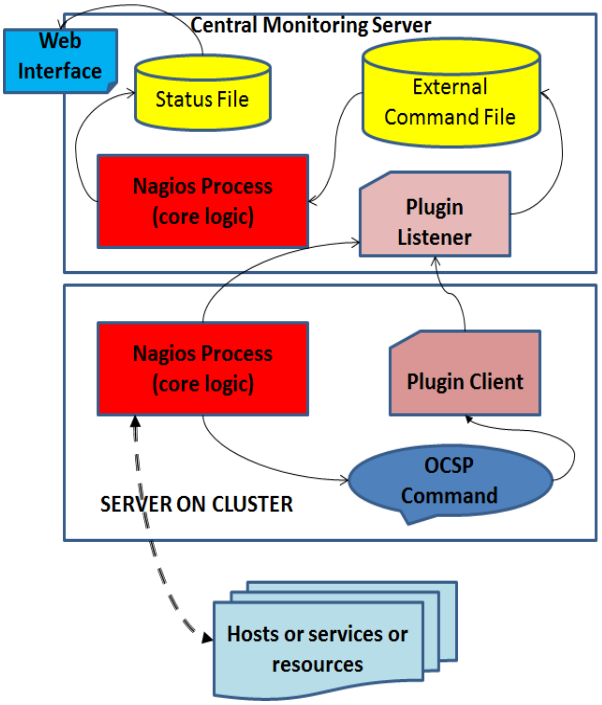

Fig 1. a) Nagios Architecture

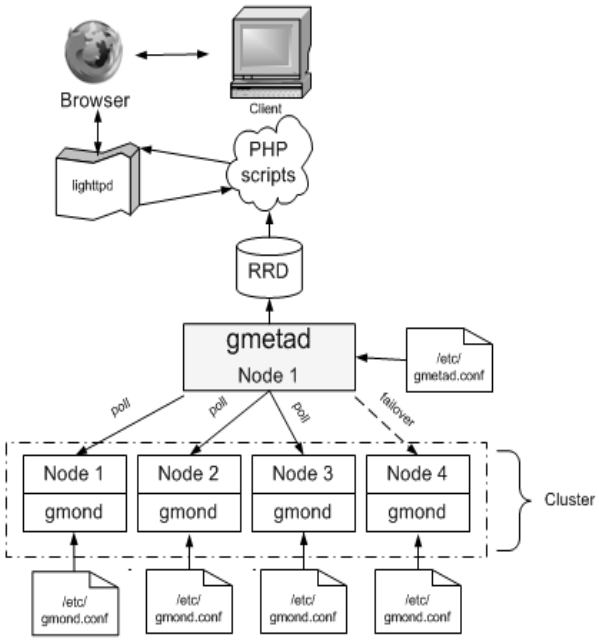

Fig. 1.b) Ganglia Architecture: (courtsey: https://my.vertica.com/docs/4.1/HTML/Master/12739.htm)

2. CMAAS FRAMEWORK (WITH NAGIOS)

We propose a framework and first present an overview of the working of the CMaaS. The figure 2 gives an overviewed idea about its working. Each host in the cluster will hold the Nagios plugin. Similarly, on every host access plugin will be there to capture access data of specific and overall fine grained services on that host. The infrastructural/QoS specific logs gathered by the Nagios plugin will be sent to the core running at monitor server along with the access logs.

More formally the access logs will be sent in an event oriented fashion whereas the QoS metrics can be sent in a time step fashion. That is, for an instance, a user logs in and starts service A the log will be formed then which is based on the event of access. The availability of that server can be checked periodically by an agent or plugin and can be logged. The monitor will consolidate the logs and keep it for recording.

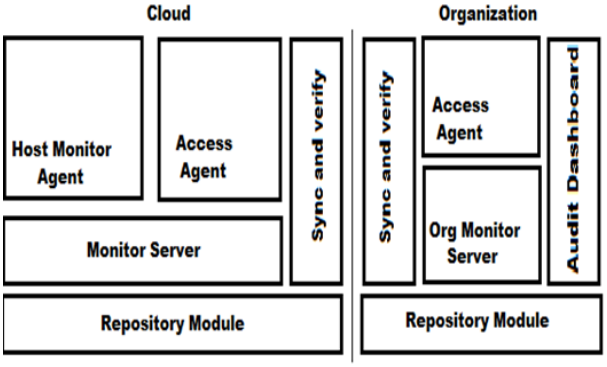

Fig 3.CMaaS Architecture

The CMaaS model was first introduced in [7] with only an overviewed idea. One can establish the overview diagram in [7] with that presented in figure 2. In the figure 2 closer aspects of Nagios or Ganglia's roles are depicted. Majorly the monitor servers at both cloud and organizational boundaries are to be holding Nagios or Ganglia cores, which will be continuously listening to the agents or plugins at hosts or access servers.

The monitors can be configured for either time step or event step fashion as shown in figure 5. The access logs generated at the access server at the organization will be rendered in event step fashion to the monitor server at organization.

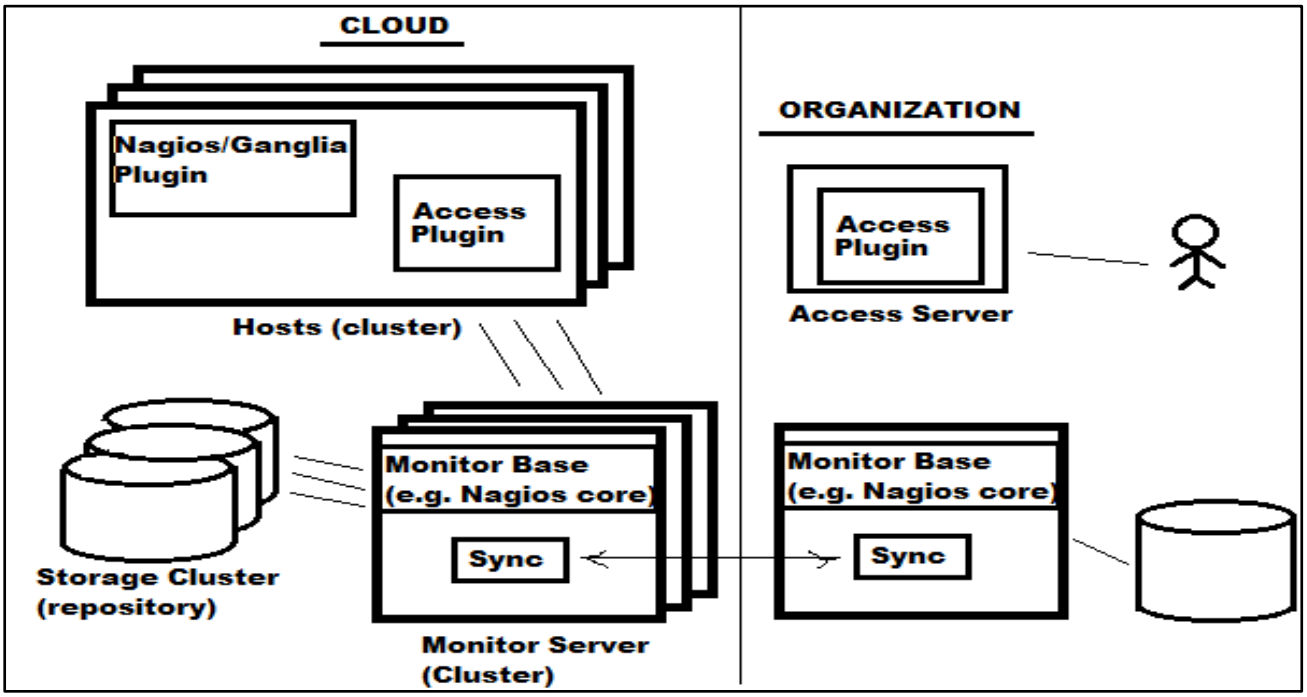

Fig 2.CMaaS Overview

The organizational monitor and the cloud monitor will synchronize in automated time step fashion or in event oriented fashion generated by the organizational admin. Organizational admin can set the metrics at both sites through the audit dashboard (figure 3). The repository module will be responsible to compact the logs and secure it as well as provide synchronization and verify sevices through the audit dashboard events.

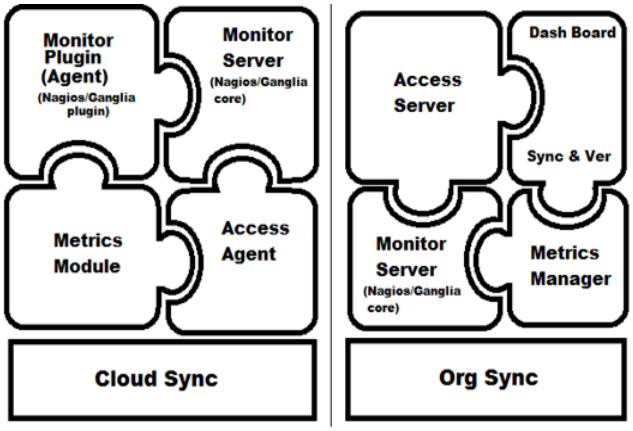

Fig 4. Interface Modules in CMaaS

The framework is designed to integrate with the Nagios. However, it can very well be set to integrate with other monitoring frameworks like Ganglia etc. The interface model is shown in figure 4. The Nagios monitor agent (plugin) will read from metrics manager module interfaced with the CMaaS agent to send logs. Similarily, the access agent reads from metrics manager and interface with CMaaS to send logs. At the organization site the access server interface with CMaaS server and both are interfaced with the dashboard, sync and verify modules. The networking layer for carrying out sync, verify etc are present as part of cloud sync and organization sync modules at every host server in hosts cluster as well as monitor server at organization.

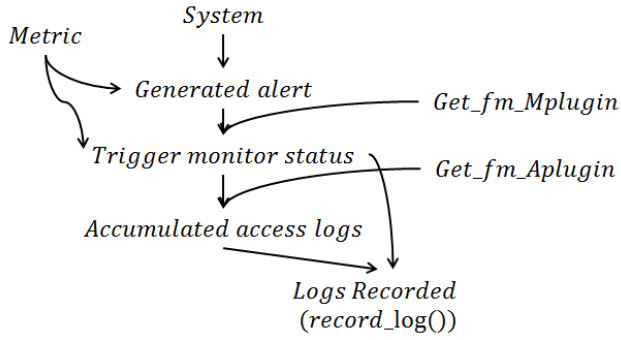

Fig. 5. a) Monitoring Flow at cloud data center core (Event Step)

Call to various methods while the functioning of the overall monitoring in CMaaS is shown in figure 5 a) b) and c). The general methods are:

$get\_fm\_Nplugin()$

$get\_fm\_Aplugin()$

$record\_log()$

$record\_log()$

$get\_recorded\_log()$

$recieve\_org\_log()$

$send\_record\_to\_cloud()$

$recieve\_cloud\_log()$

$Store\_log()$

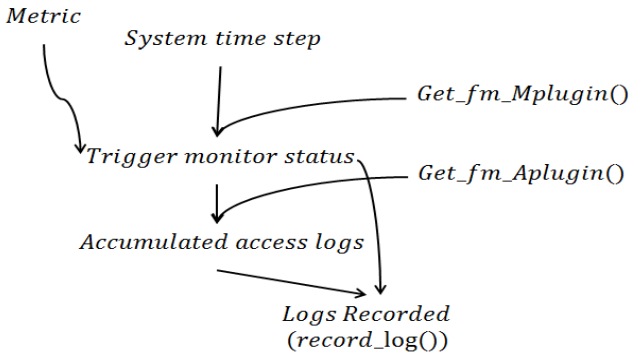

Fig. 5. b) Monitoring Flow at cloud data center core (Time Step)

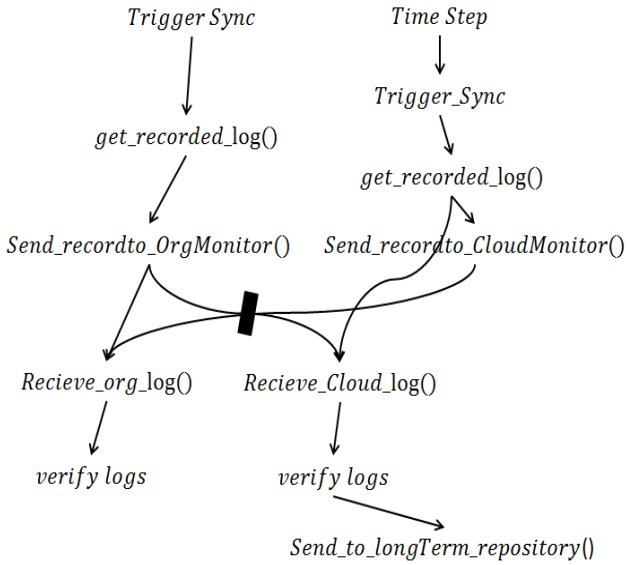

Fig. 5. c) Time Step Sync and verify for audit (Black rectangle is the line to mark communication channel)

The exchange of recorded logs will be done in a full SSL model in the dialogues shown hereby:

For a time period from i to j:

1. At cloud at t_{j+1} : $Hash\_Encrypt(T_p + \sum_{t_i}^{t_j} CD + \sum_{t_i}^{t_j} CA) \rightarrow C_{dig}$ (digital signature)
2. At organization at t_{j+1} : $Hash\_Encrypt(T_u + \sum_{t_i}^{t_j} UD + \sum_{t_i}^{t_j} UA) \rightarrow U_{dig}$ (digital signature)
3. At organization at t_{j+2} : $Compare(C_{dig}, U_{dig}) \rightarrow$ Verification

Where,

$T_p \rightarrow$ Task metric at provider.

$CD_{t_j} \rightarrow$ Data Center monitored log.

$CA_{t_i \; to \; t_j} \rightarrow$ Cloud Access monitored logs.

$T_u \rightarrow$ Task metric at user organization.

$UA_{t_i \; to \; t_j} \rightarrow$ Access monitored logs at organization.

$UD_{t_i \; to \; t_j} \rightarrow$ Data center monitored logs synched with User organization.

3. CONCLUSION AND FUTURE WORK

The aspect of having a complete monitoring in cloud computing services is desirable. However, with the present set of available monitoring over cloud is not complete in the sense that it doesn't support fine grained access monitoring. A complete framework that can be incorporated with certain bench mark open source monitoring tools for cloud computing like, Nagios is presented hereby. The framework can be adjusted for the other monitoring tools like Ganglia. The framework can be implemented to have a centralized access management with fine grained access monitoring on clouds. The framework is mostly presented through the set of diagrams. However, we will be shortly coming up with a prototype which can demonstrate the possibilities and extent of implement-ability of the above framework.

4. REFERENCES

[1] "OpenStack: An Overview", http://www.openstack.org/downloads/openstack-overview-datasheet.pdf, retrieved Feb 2014.

[2] "OpenNebula2.0 Architecture", http://archives.opennebula.org/documentation:archives:rel2.0:architecture, retrieved march 2014.

[3] "NagiosXI architecture", http://www.nagios.com/products/nagiosxi/architecture, retrieved march 2014

[4] Matthew L. Massie, Brent N. Chun , David E. Culler "The Ganglia Distributed Monitoring System: Design, Implementation, and Experience", http://ganglia.sourceforge.net/talks/parallel_computing/ganglia-twocol.pdf, retrieved march 2014.

[5] "Nagios Architecture", http://www.onaxer.com/2010/01/24/nagios-architecture/, retrieved march 2014.

[6] "Ganglia architecture", https://my.vertica.com/docs/4.1/HTML/Master/12739.htm, retrieved march 2014.

[7] Ajay Prasad, Prasun Chakrabarty, "Centralized Access Management and Monitoring as a Service in Cloud Environments-A Critical Study", Computer and Information Science (CIS), Volume 6, No.2, 126-123, 2013.