Trustworthy Resource Scheduling using Openstack in Cloud

Yogesh V. Jilhawar
Department of Information Technology
Pune Institute of Computer Technology
Pune, India.
yvj9391@gmail.com

Dr. Emmanuel M.
Head and Professor
Department of Information Technology
Pune Institute of Computer Technology
Pune, India.
emman2001@gmail.com

***Abstract*—Now a days, we have number of cloud infrastructure as a service (IaaS) frameworks. The overall function of these systems is to manage the provisioning of virtual machines for a cloud which provides IaaS. All these open source platforms provide an alternative solution for those who do not wants to use a service provided by commercial cloud service providers. cloud actors like users, administrators and developers have to take decision about suitable environment for them.**

Having trust on the underlying infrastructure is vital from the point of user. Also, at the same time the underlying instance must not perform any malicious activity. In order to detect malicious behavior of such an instance, we provide a chain of trust starting with hardware chip followed by trusted BIOS. Infrastructure properties are calculated and from calculated properties malicious behavior of instance is detected. If found, that instance is migrated to another computing node.

Keywords-BIOS, IaaS, instance, malicious, virtual machine.

I. INTRODUCTION

Cloud is an emblematic word for internet. So cloud computing refers to a type of computing which is based on internet. Cloud computing is concerned with applications and services that run on a distributed network using resources that are virtualized and accessed through common Internet protocols and networking standards. It creates illusion that resources are not having any upper bound and details of the physical systems on which software runs are kept hidden from the user. Cloud computing is an computing environment in which computing done by one party can be used by another party [4] and internet is used to access the computing power and resources like database [7]. Cloud computing is a new trend that keeps data into large data centers which is far away from accessing place. In cloud based environment, there is no need for customers to pay for infrastructure, installation and maintenance. Cloud computing models are broadly classified into two, deployment model and service model. Formal model briefs about the location and purpose of the cloud, examples includes public, private hybrid and community clouds. Later one briefs about the type of service the service provider is offering, examples include software as a service, platform as a service and infrastructure as a service [2]. Physical resource are Computer, disk, database, Bandwidth, Processor, scientific instruments, network and the logical resources are Execution, monitoring, communicate application etc. The cloud environment raises a number of challenges. Two parties in cloud environments, cloud suppliers and cloud consumers, has different goals; suppliers aims at maximizing the revenue by achieving high resource utilization, while consumers aims at minimizing expenses considering their performance requirements. But, it is difficult to allocate resources such that both parties get their optimal benefits because of the lack of information sharing between them [4]. However the increasing heterogeneity and variability of the environment leads to even harder challenges for both parties.

With increasing need for cloud to provide multi-tenancy many platforms have evolved and are used by many organizations. Openstack is one of these and we used it for our project. Other platforms include Eucalyptus [10], open nebula [11], nimbus [12].

Openstack is an open source cloud computing platform developed commonly by rackspace and NASA to have private cloud for organizations. Openstack is also a collection of open source components to develop both private and public cloud. Current version of openstack is Icehouse which has 10 different sub-components viz. Nova, neutron, swift, cinder, horizon, keystone, glance, ceilometer, heat and trove. Nova service provides the management of lifecycle of compute instances in an OpenStack environment. Nova is responsible for spawning, scheduling and deactivating virtual machines on demand [6].

Neutron provides Network-Connectivity-as-a-Service for OpenStack Compute. Neutron also provides users an API that defines networks and the attachments between them. In order to support many popular networking technologies and vendors open stack offers a pluggable architecture. Openstack is quite difficult to deploy since it needs to configure number of modules. Each new version of openstack [9] comes after every six to seven months. Providing storage for VM images is becoming vital in case of providing Infrastructure as a service. Openstack includes a sophisticated storage system called as Swift. Images can be stored in POSIX file system transferred using ssh or in swift transferred using http/s.

OpenStack supports two modes of managing networks for virtual machines: vLAN networking and flat networking. vLAN based networking sets up vLANs for systems by using vLAN capable managed switch. Flat Networking connects multiple compute hosts together with the help of Linux Ethernet bridging. Open stack supports most of the hypervisors including KVM, XEN [13], VMWare, Vsphere, LXC, UML and MS HyperV which makes openstack an attractive choice for experimenting with different hypervisors. For authentication open stack uses X509 credentials and LDAP.

II. RELATED WORK

This section introduces an overview of papers related to trustworthy scheduling and open stack scheduler. First paper gives resource scheduler which provides trustworthiness. Second paper briefs about scheduling depends on Service level agreement. Last paper introduces scheduling of virtual machines.

Imad abbadi *et al*. [1] proposed a trusty cloud scheduler which considers user requirements as well as infrastructure properties. They assures users that their virtual resources are hosted using physical resources that match their requirements. Users are not involved in assigning the cloud infrastructure. They provided a prototype which implements the cloud scheduler. This paper specifically focuses on providing the scheduler with trustworthy input about the trust status of the cloud infrastructure. This paper covers one of the most important properties which is about measuring the trust status of the cloud infrastructure, and enabling users to define their minimal acceptable level of trust.

Jordi Vilaplana *et al*. [2] focuses on the scalability problem in cloud-based systems when changing the computing requirements, this is, when we have a high degree of variability in requesting service. They studied a specific scenario for web-based application deployed in a cloud system, where the number of requests can change with time. This paper provides assurance about SLA (Service-Level Agreement)in scalable clouds . They proposed an architecture that balances the load between different computing virtual machines. For that purpose they monitored the system to draw conclusion about when to create or terminate virtual machines. Results obtained by implementing the proposed architecture proves the applicability of their proposal for guaranteeing SLA in a real cloud framework.

Khaled M. Khan *et al*. [3] discusses factors that affect consumers' trust in the cloud and some of the emerging technologies that could be used to bring trust in the cloud including enabling more jurisdiction over the consumers' data through provision of remote access control, transparency in the security capabilities of the providers, independent certification of cloud services for security properties and capabilities and the use of private enclaves.

Brian Hay *et al*. [4] suggests some technical mechanisms including encrypted communication channels and computation on encrypted data as ways of addressing some of the trust challenges.

Gourav sharma *et al*. [5] has discussed various steps in OpenStack installation on commodity hardware. They started with installing Ubuntu 12.04 LTS on all nodes and progressed to installation of various packages on Controller and Compute nodes. They have also described the required database structure for OpenStack installation. They have provided scripts automating this process. They have successfully deployed the cloud in two scenarios. First, They have deployed cloud over machines connected by a closed network of few machines. Then they have used the existing network infrastructure, which allows to potentially use geographically separated nodes.

III. PROCESSING MODULES

A. Computing node chain of trust

This module consist of establishing a chain of trust at computing node. Building the RCoT(Resource Chain of Trust) of a computing node has the various resources upon which trust can be established. Resources includes hardware chip, BIOS, Grub, Kernel etc.

B. Computing node attestation

This module includes the trust establishment protocol. Controller node sends an attestation request to computing node. Computing node collects the integrity measurement log as recorded by IMA and generates the IR following signed PCR values.

C. Processing module

Access Control as a Service is a scheduler which communicates with DC-S and nova database to allocate one of the resources for the user request. It is present at the controller node.

At computing node, it collects the integrity measurement logs as recorded by the IMA and from that generates IR(Integrity report). It sends IR report and signed PCR (platform configuration register)values to controller node.

At controller node, it collects and verifies IR report and signed PCR values send by the computing node.

D. Instance migration

This module mainly focuses on migration of currently going on instance on some computing node to another one, as per values given by computing node because of detection of malicious activity performed by that particular computing node. When such an activity detected controller node tries to find out another computing node which will satisfy the current instance request and migrate to that node.

Work mainly focuses on providing a trustworthy scheduling algorithm that can automatically manage the cloud infrastructure by considering both user requirements and infrastructure properties and policies. Also develop the required trustworthy software agents which automatically manage the collection of the properties of physical resources.

Software agents will be present at both controller node and at computing node. Agent present at controller node have to collect report send by the agent at compute node. From the report, trust status of the instance is found out and decision is taken whether to load instance on that particular computing node or not. Similarly, agent present at computing node will calculate the infrastructure properties by reading values from PCR registers and then generated IR(Integrity report) report will be forwarded to agent present at controller node.

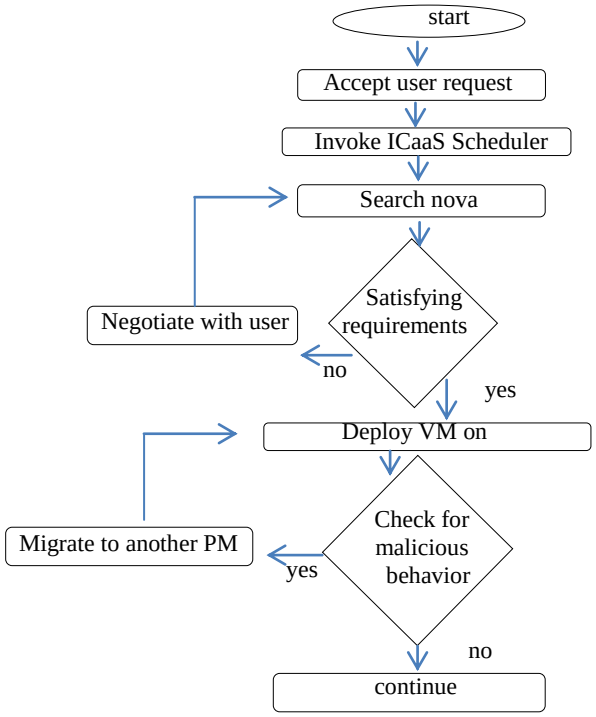

Figure 1: flow diagram of instance allocation.

Request from user is accepted at controller node.Appropriate computing node is selected and instance is launched on that node by the scheduler for that scheduler checks the nova database. If requirements match with the any computing node , instance is launched. If launched instance performs any malicious activity then then properties of underlying infrastructure changes as on recorded in the PCR registers on TPM chip. Controller node get notification about this. Scheduler on controller node search for another healthy computing node and migrates that instance on that node.

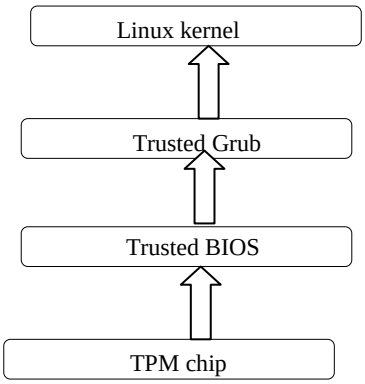

Figure 2: Resource chain of Trust at computing node.

IV. TRUST ESTABLISHMENT

A. *TPM:-*

TPM (Trusted Platform Module) is a computer chip (microcontroller) that can securely store values used to authenticate the platform (your PC or laptop). These things can include passwords, certificates, or encryption keys. A TPM is mainly used to store infrastructure measurements which ensures trustworthiness of underlying platform. Authentication (ensuring that the platform can prove that it is what it claims to be) and attestation (a process helping to prove that a platform is trustworthy and has not been breached) are important steps to ensure computing is free from danger in all environments. TPM version 1.2 is used for the project.

B. *Trusted Grub :-*

Trusted GRUB extends the GRUB boot loader with TCG support. This makes it possible to provide a secure bootstrap architecture. The whole boot process is measured and – with the help of a Trusted Platform Module (TPM) - the system integrity can be checked. Trusted GRUB is an enhancement to the GNU GRUB boot loader, which supports Authenticated Boot as well as Secure Boot using the Trusted Platform Module.

C. *IMA :-*

Integrity Measurement Architecture (IMA) is an IBM project which provides runtime measurements and attestation for Linux. The newly released Linux kernel 2.6.30 includes Integrity Measurement Architecture (IMA) in the main file system! IMA is a platform for trusted computing and it was listed by Computer world as one of the top 5 new features of 2.6.30.

V. PROTOCOL

Protocol : computing node registration protocol

Computing node(Ci) sends a registration request to controller node (M) as follows:

1. Ci sends a request to its TPM to create an AIK key pair using the command TPM_CreateAIK. The TPM would then generate an AIK key pair.
2. The generated private section of the key pair never leaves the TPM, and the corresponding public section of the key pair is signed by the TPM Endorsement Key (EK). The EK is protected by the TPM, and never leaves it.
3. Ci then sends a registration request to M. The request is associated with the EK certificate, the AIK public key and other parameters.

M certifies AIK_i as follows :

1. M verifies Cert(EK_i). If the verification succeeds, M generates a specific-AIK certificate for Ci and a unique ID, CID_i. It then sends the result to Ci.

VI. CONCLUSION

In this paper, we have seen open stack as the one of the best cloud operating system in regard of scalability. Providing trust about computing node is vital from the point of view of cost for maintenance if any vulnerability found. Also, at the same time underlying instance shouldn't perform any malicious activity. Providing security by having resource chain of trust, starting from hardware chip, is important from the point of view of making underlying infrastructure trustworthy in the era of cloud computing.

REFERENCES

[1] Imad M. Abbadi and AnbangRuan, "towards trustworthy resource scheduling in clouds",IEEE transaction on information forensics and security,vol.8 no.6, JUNE2013.

[2] JordiVilaplana, FrancescSolsona, Jordi Mateo, and Ivan Teixido,"SLA-aware load balancing in a web-based cloud system over openstack",Springer International Publishing Switzerland pp-281–293, 2014.

[3] K. M. Khan and Q. M. Malluhi, "Establishing trust in cloud computing", IT Professional, vol. 12, no. 5, pp. 20–27, Sep. 2010.

[4] B. Hay, K. L. Nance, and M. Bishop, "Storm clouds rising: Security challenges for IaaS cloud computing",in Proc. HICSS IEEE Comput. Soc., pp. 1–7 2011.

[5] Diplav Dongre, gourav Sharma " Scalable cloud deployment on commodity hardware using open stack" advanced computing,networking and informatics-volume2, springer 2014.

[6] I. M. Llorente, R. Moreno-Vozmediano, and R. S. Montero, "Cloud computing for on-demand grid resource provisioning," Advances in Parallel Computing, vol. 18,pp. 177-191, 2009.

[7] "cloud computing bible" by Barrie sosinsky ISBN: 978-0-470-90356-8.

[8] B. P. Rimal, E. Choi, and I. Lumb. A Taxonomy and Survey of Cloud Computing Systems. Fifth International Joint Conference on INC, IMS and IDC, pages 44–51, 2009.

[9] Openstack home page, http://www.openstack.org Nov2014.

[10] Eucalyptus Home Page, http://www.eucalyptus.com/ Nov2014.

[11] Open Nebula Home Page, http://www.opennebula.org/ Nov2014.

[12] Nimbus Home Page. http://www.nimbusproject.org/. Nov2014.

[13] Amazon EC2 Spot Instances. http://aws.amazon.com/ec2/spot-instances/ 2014