IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. VI (Mar-Apr. 2014), PP 95-101
www.iosrjournals.org

An Approach to Steadfast and Reliable Resource Scheduling in Cloud Environment

A. Aalan Babu, S. Roselin Mary
P.G Scholar, Department of Computer Science and Engineering, Anand Institute of Higher Technology, Chennai, India.
Associate Professor, Department of Computer Science and Engineering, Anand Institute of Higher Technology, Chennai, India.

Abstract: *In cloud computing, resource allocation and scheduling are the necessary requirements which helps the cloud users to access the resources that are available. Applications implemented in the cloud often required to interact amongst themselves and, in case, rarely depend on other implemented applications. Present cloud schedulers do not consider the infrastructure properties of cloud during resource allocation and scheduling process. In this paper, we propose a reliable resource scheduling algorithm that can automatically manage the cloud infrastructure by considering both user requirements and infrastructure properties and policies. A new prototype is presented based on Openstack and it implements the proposed cloud scheduler. Our work specifically focuses on providing the cloud scheduler with trustworthy input that expose the trust status of cloud infrastructure and it also further establish the foundations of planned future work to cover other properties. It further provides an implementation of the past work on trust management of cloud which provides the cloud scheduler with input about the accurate trust status of that cloud infrastructure.*
Keywords: *Cloud; Resource Scheduling; Cloud Scheduler; Openstack; Trust status.*

I. INTRODUCTION

Cloud infrastructure is generally complex and heterogeneous in nature which provides numerous components by various vendors. Applications implemented in the cloud often required to interact amongst themselves and, in case, rarely depend on other implemented applications. The rapid complexity of the infrastructure and application dependency leads an environment that requires management and raises security and privacy problems. The central component that restricts and controls the allocation of virtual resources for a cloud infrastructure's physical resources is known as the cloud scheduler. Presently available schedulers do not consider users' security and privacy needs; neither do they consider the properties of the entire cloud infrastructure. So, a cloud scheduler should consider the necessary application performance requirements and also the user's security and privacy requirements.

This paper proposes a reliable resource scheduling algorithm that automatically manages the cloud infrastructure by considering both user requirements and infrastructure properties along with its individual policies. It also further develops the required trustworthy software agents that automatically manage the collection of the specific properties of physical resources. Maintaining a trustworthy and specific copy of the infrastructure properties and user's requirements is critical for the absolute operation of the scheduler. This paper mainly focuses on providing the cloud scheduler with trustworthy input about the trust status of the cloud infrastructure which further establishes the foundations of planned future work.

Open Stack refers to its specific cloud scheduler component using the term "nova-scheduler" and it identifies the cloud scheduler as the most complicated component to develop the significant effort still remains to have an appropriate cloud scheduler. In order to develop the trustworthy resource scheduler component, it is necessary to have the exact understanding of how clouds are managed and how they work in the environment. From this, it is finally concluded that establishment of trust in clouds requires two mutually dependent elements:

- Appropriate trustworthy mechanisms and its tools to help cloud service providers activate the process of managing, maintaining, and securing the cloud infrastructure; and
- Implementing methods for cloud users and cloud providers for establishing trust in the operation of cloud infrastructure.

This includes mainly supporting the cloud infrastructure with trustworthy and self-managed services that manages the entire cloud infrastructure. The automated self-managed services must provide the cloud computing environment with additional capabilities and more features. The example for additional capabilities are scale per use, hiding the cloud infrastructure's complexity, automated true reliability, current availability, existing scalability, dependability, and resilience that considers the cloud users' security and privacy requirements by the properties.

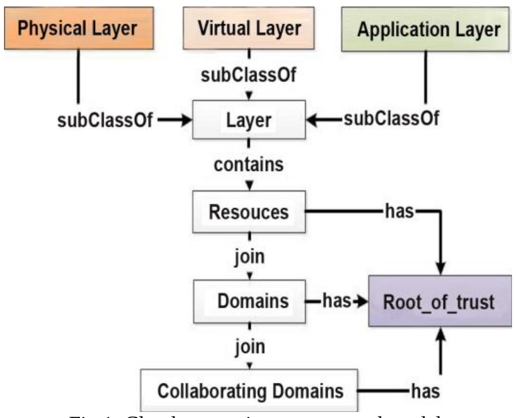

Fig 1. Cloud computing - conceptual model.

A. *Cloud infrastructure overview*

Generally, the cloud environment is composed of numerous resources, that are categorized depending their types and implementations across the overall cloud infrastructure. A resource in cloud is a conceptual entity that provides enormous services to other entities.

Basically, cloud environment consists of mainly three intersecting layers as follows:

1) *Physical Layer* - This layer denotes the key physical resources and their interactions that construct a cloud's physical infrastructure. Physical layer resources are assisted to serve the Virtual Layer;
2) *Virtual Layer* - This layer represents the key virtual resources, which are directly assisted by the Physical Layer; and
3) *Application Layer* - This layer executes the applications of the cloud user which are assisted by the Virtual Layer resources.

Fig.1 provides the conceptual model that explores an entity layer which is represented as the parent of three cloud layers (i.e., physical, virtual, and application layers). From the abstract level, a layer constitutes the resources which join the domains (i.e., physical domains, virtual domains, and application domains). A domain reflects the container which consists of similar related resources. Each domain resources are maintained by following the domain defined policy. Domains that need to collaborate among themselves within a layer join a collaborating domain (i.e., physical collaborating domains, virtual collaborating domains, and application collaborating domains). A Collaborating domain dominates the interaction between its fellow domains using a well-defined policy.

The specific nature of resources, domains, Collaborating domains, and their identical policies are layer dependent. Domains and Collaborating domains are the main concepts that help in managing the entire cloud infrastructure, and maintaining resource distribution and mutual coordination in normal operations and performances. Each of the identified cloud entities includes a root of trust separately that helps establishing trust in clouds.

B. *Virtual Control Center*

At present, there are many tools for performing a cloud's virtual resources, e.g., vCenter and OpenStack. For our convenience, we can name those tools using a common name "Virtual Control Centre (VCC)", which is a cloud device that manages the overall virtual resources along with their collaborations with physical resources by using a specific set of software agents. The VCC plays a vital role in providing the cloud's automated and self-managed services, which are most commonly provided manually during the time of execution.

C. *Cloud chains of trust*

One of the major key properties of a cloud infrastructure is the trustworthiness with attention to the managing users' virtual resources on physical resources which was agreed in service level agreement (SLA). Estimating the trust levels of clouds is not only favourable to cloud users, but also guides the cloud providers in understanding how their infrastructure is operated and managed. However, this is a highly complicated problem to deal with considering the effective nature and numerous resources of the cloud infrastructure. Some of the proposed schemes are analyzed here; and some of those focus on scaling the trustworthiness of the overall cloud infrastructure while others attempt to build a chain of trust with the resources available at a specific point in time duration. This problem has been analyzed, and found that it is not practical to estimate the trustworthiness of the overall cloud infrastructure (considering its huge complexity). Our method is based on focusing the cloud infrastructure and estimating the trustworthiness of each segment independently. The boundaries of each individual segment are based on how the infrastructure is maintained in general. Specifically, the boundaries are coordinated using the domains and collaborating domains concept which are framed earlier. The concept of cloud chain of trust is introduced which provides a single chain of trust by representing a collective group of entities. This is very essential in clouds as many entities exist together as a composition of multiple entities.

D. ***CLOUD Requirements***

Context

The context of a system refers to the connections and relationships between the user and cloud environment. It defines the resource scheduling process in the cloud environment.

Authentication

It is necessary that the interface defined between the cloud user, the system and the admin has to provide authentication. Authentication is provided by accessing the user name and password and this helps the application in protecting sensitive information from the users.

Authorization

The access to the application will be granted only to those users who have gained the approval of the admin. This becomes possible only after each user creates a user login and password. Authorization determines the unique privileges granted to the admin approved users.

Performance

The performance of the processed resource scheduling is to be measured. So it is done after the resource scheduling is finished. Also the requirements determine the resources required, response time, transaction rates, throughput and everything that deals with the performance of the system.

Security and Privacy

The Security side describes the need to control access to the data. This includes controlling who may view and alter the application data. List of the security requirements using the following criteria:

- Loss of node should be avoided.
- Provide access only by user roles.

Reliability

Reliability is the probability that the system works correctly and completely without being aborted. Reliability is also the probability of how often the software fails.

Usability

The usability requirement specifies how easy the system must be to use. Usability is a non-functional requirement, because in its essence it doesn't specify parts of the system functionality. All the usability requirements are gathered at the start of each process.

II. HIGH LEVEL ARCHITECTURE

The high level architecture is presented in Fig.2 which outlines the main entities and the general design of our scheme framework. We use the OpenStack controller node (i.e., the VCC) and an OpenStack nova-compute (i.e., a computing node in the physical layer). The computing node runs an interpreter which manages a set of cloud VMs. The VCC receives two main inputs: cloud user requirements and cloud infrastructure properties and it manages the user virtual resources that are based on these specific inputs. In this section we focus the components of Open-Stack.

A. ***Nova-Api***

Nova-api is a collection of command line tools and graphical interface that are used by cloud users while managing their resources in the cloud environment, and are also used by cloud administrators and schedulers when managing the overall cloud virtual infrastructure. The cloud virtual infrastructure is well organized and maintained. Examples of those properties include: physical resources reliability and connectivity, physical resources distribution across the entire cloud infrastructure and resources clustering and grouping. The cloud user properties include specialized requirements, service level agreements, and security and privacy necessity requirements.

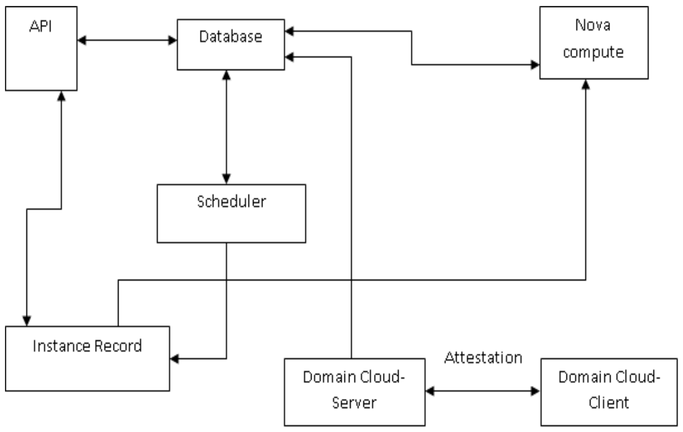

Fig 2. High level architecture

B. ***Nova-Database***

Nova-database is composed of mainly databases which include the details of the cloud components, cloud users, cloud projects and cloud security properties. The nova-database is extended in various directions just to visualize the taxonomy of clouds, and also to cover the user security requirements and cloud infrastructure properties. Compute_nodes is an already existing nova-database table that holds all records in showcasing computing resources at the top physical layer. The nova-database table is updated by adding the following fields: RCoT (Physical) and cloud security properties that holds values of the cloud computing resources security details.

C. ***Nova-Scheduler***

Nova-scheduler manages the hosting of cloud VMs at physical resources by considering cloud user requirements and cloud infrastructure properties. Present implementation of nova-scheduler does not deal with the entire cloud infrastructure, and also they does not consider the cloud user and overall infrastructure properties. According to OpenStack authorization, nova-scheduler is still inexperienced and huge efforts are still enforced to promote it. We implement a modern scheduler algorithm, ACaaS (Access Control as a Service). It considers the cloud taxonomy which is reviewed earlier, and it selects a key physical resource that has properties which can best match with the requested cloud user requirements. Also it ensures that the cloud user requirements are constantly maintained. The proposed ACaaS scheduler hooks up with the following cloud software agents.

- The cloud client agent, called DC-C functions at each individual OpenStack computing node. It estimates the computing node's RCoT, constantly looks on the status of the computing node, and then forwards the result to the DC-S, and also operates cloud domains and collaborating domain members placed on protocols provided by the DC-S.
- The cloud server agent, called DC-S functions at OpenStack domain cloud controller and controls the OpenStack components by assuring that they perform the cloud only when they reach the trust status. It also attests with the DC-C trustworthiness when the cloud computing node joins with its resource physical domain.

III. EXISTING WORKS

The subject of building trust in the cloud environment has been examined by many authors (e.g., [3], [4], [5], [6]). Part of the discussion has been focused on specific reasons whether to "trust the cloud environment" or not to. The work in [4], [5] focuses on identifying the properties for establishing trust in the cloud. Sheng Di and Cho-Li Wang [7] proposed a novel resource allocation algorithm for cloud system that supports VM-multiplexing technology, aiming to minimize user's payment on his/her task and also endeavour to guarantee its execution deadline. It can be proved that the output of algorithm is optimal based on the Karush-Kuhn-Tucker (KKT) condition, which means any other solutions would definitely cause larger payment cost. Frank Doelitzscher et al. [2] discussed about the prototype demonstration of the Security Audit as a Service (SAaaS) architecture, a cloud audit system which aims to increase trust in cloud infrastructures by introducing more transparency to user and cloud provider on what is happening in the cloud. Sheikh Mahbub Habib et al. [6] provides an overview of Trust Management system architecture for cloud computing marketplace. It aims at supporting customers to identify trustworthy services providers as well as trustworthy service providers to stand out.

Imad M. Abbadi [4] analyzed the most important clouds' properties, which enable different interested parties to assess the operational trust of a cloud provider for delivering services. The assessment of operational trust enables cloud users, auditors, collaborating cloud providers, and others to compare multiple cloud providers and decide on the best value for money. It also enables cloud providers to understand what could be done to improve the level of services. Brian Hay et al. [1] focused on the complex security challenges in Infrastructure as a Service (IaaS) based cloud computing. While not exhaustive, it identifies some technological and legal issues and concerns from the perspectives of identified stakeholders, and further suggests some future directions for security research and development in advancing the security needs. Jemal Abawajy [5] presented a fully distributed framework that enable interested parties determine the trustworthiness of federated cloud computing entities. Soren Bleikertz et al. [8] presented a novel approach in the security assessment of the end-user configuration of multi-tier architectures that is deployed on infrastructure clouds such as Amazon EC2. In order to perform this assessment for the currently deployed configuration, the process of extracting the configuration is automated using the Amazon API. Also it has been focused on the reachability and vulnerability of services in the virtual infrastructure, and presented a way for the visualization and automated analysis based on reachability and attack graphs. Imad M. Abbadi & Cornelius Namiluko [3] focused on deriving some of the challenges for trust establishment in cloud computing. It is done by providing a conceptual model of cloud infrastructure, and then frames the dynamic nature of cloud based on the provided model. It is impossible to attest to the overall cloud infrastructure taking on its huge and shared resources. This work needs resources, when collaborating with other available resources, to attest to their trustworthiness level.

Few papers present the use of TPM in cloud environment for remote attestation process ([6], [2]). The work done here reflects a remote attestation method based upon reputation based systems and appropriate TCG remote attestation methods. The cloud trust measurements are linked with an individual timestamps and would need to be revised after it concludes. The propagation of such measurements among resources forms a chain of trust. The relations among entities in clouds are analyzed mainly due to the potential behaviour of those entities, and the entities of chain of trust need frequent reviews.

IV. PROPOSED MODEL

In the proposed model, we newly present the prototype that is built on Open Stack. The afforded prototype applies the newly presented cloud scheduler. It also yields an implementation of the past work on cloud trust management that yields the scheduler with input about the current trust status of the overall cloud infrastructure**.** The technical functions implied to calculate the cloud compositional chains of trust serve various levels of transparency based on the cloud user type (IaaS, PaaS, or SaaS). The proposed model explores the performance of the remote attestation process and the secure scheduling method.

1) *Remote Attestation process:* The presented prototype implements the process of remote attestation that is managed by the DC-S. The implementation of the trusted channel, when sealed keys are loaded into memory, needs a small Trusted Computing Base (TCB). The TCB should carry out a strict approach to the memory area that locates the key. But having a large TCB, can lead to leaking the key from memory without reflecting on the cloud platform trust status. Hence implementing a small TCB is a serious problem, especially, taking on the complication and scalability of the specific hosting cloud system. Also as an attempt to look on the impact of this threat, we impose periodic verifications in the presented prototype which maintains the security properties of a cloud computing node up to date. This is implemented by associating a timer with every individual computing node. Reattestation is applied at once, whenever the timer expires. Untrustable cloud computing nodes discovered by the reattestation will be at once removed from the database and it need to re-enter in the system for further use.

2) Secure Scheduling method: In general, computing nodes are standardized into physical domains. Such standardization would be based on the specific properties of every individual computing node (i.e., security and privacy) that enables it to perform the requirements of the domain. Users are able to specify their desired properties of computing nodes that would host their VMs. In the proposed prototype, users must analyze their respective hosting environment using the afforded sets of white-lists. A cloud computing node’s white-list is determined in assistance with its specific properties that get attested while combining a domain. Appropriate changes on the properties of a computing node require the modification of the corresponding record in the white-list database. In our presented prototype, some users need properties that express the entries in database. The ACaaS cloud scheduler implements each VM on a computing node which has the same identical properties as the one demanded by the user of that VM. The ACaaS cloud scheduler, with association with the DC-S and the DC-C, often inspects the flexibility of such properties.

The presented prototype permits cloud users to identify their expected trust level at the physical resources hosting their virtual resources without the need to get involved with overall cloud infrastructure complexity. It takes on the critical factors that have not been considered in cloud commercial schedulers such as, considering the entire cloud infrastructure and computing nodes’ trust status.

Algorithm

The proposed algorithm in this paper is the Reliable Resource Scheduling Algorithm. This algorithm automatically manages the cloud infrastructure by considering both user requirements and infrastructure properties and policies.

Remote Attestation Process

The attestation protocol works as follows. Every computing node (C_i) is identified by its AIK (Attestation Identity Key). The AIK is certified by the cloud controller VCC (M) as it covers the Privacy-CA. When a new computing node is added to the cloud infrastructure it must be first registered at the VCC which then certifies its AIK. Only registered computing nodes can connect to the VCC as their certified AIKs cannot be forged and AIKs can only be used inside the genuine TPM that generates them. The registration steps of C_i at M is outlined in Protocol 1. Whenever a computing node sends a request to connect to the VCC a trust establishment protocol is executed which is outlined in Protocol 2.

Protocol 1: Computing Node Registration Protocol.

1) C_i sends a registration request to M as follows . First, C_i 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. The generated private part of the key pair never leaves the TPM, and the Corresponding public part of the key pair is signed by the TPM Endorsement Key (EK). The EK is protected by the TPM, and never leaves it. C_i then sends a registration request to M . The request is associated with the EK certificate, the AIK public key and other parameters.

   $C_i \rightarrow M : Cert(K_{EK_i}), \{K_{AIK_i}\}_{K_{EK_i}^{-1}}$

2) M certifies AIK_i as follows: M verifies (EK_i) . If the verification succeeds, M generates a specific-AIK certificate for C_i and a unique ID, CID_i. It then sends the result to C_i.

   $M \rightarrow C_i : \{Cert(K_{AIK_i}), CID_i\}_{K_M^{-1}}, Cert(K_M)$

Protocol 2: Trust Establishment Protocol.

1) M Sends an attestation request to C_i. The request includes a nonce N_a.
2) C_i would then report an attestation ticket to M as follows. C_i sends its values, and the measurement log IR back to M, together with N_a . These are signed using the C_i's AIK.

   $C_i \rightarrow M : \{N_a, \{PCR\}, IR\}_{K_{AIK_i}^{-1}}$

3) M then verifies the message sent by C_i as follows. It verifies the AIK_i signature and N_a matches the sent nonce. If the verification succeeds, M examines the consistency of PCR and IR, and then determines the properties of C_i based on the value of IR.

Reliable Resource Scheduling Algorithm

1) Computing node (Resource) sends a registration request to its TPM to create an AIK (Attestation Identity Key) key pair using the command TPM-CreateAIK.
2) After the registration process, the AIK key will be generated.
3) The TPM would then generate an AIK key pair.
4) The corresponding public part of the key pair is signed by the TPM Endorsement Key (EK).
5) The EK is protected by the TPM, and never leaves it. Then sends a registration request to Cloud Machine.
6) The request is associated with the EK certificate, the AIK public key and other parameters.
7) Then Computing node sends Certificate to cloud machine. If certificate matches then Cloud Machine generate another AIK certificate key.
8) Both the Attestation Identity Key and Endorsement Key are compared.
9) If both the keys are matched, then the TPM provides accessibility of resources.

TABLE I. NOTATIONS

Notations	**Description**
TPM	Trusted Platform Module
ACaaS	Access Control as a Service
SLA	Service Level Agreement
VCC	Virtual Control Center
AIK	Attestation Identity Key
EK	Endorsement Key

The above table shows the notations used in this paper.

V. CONCLUSION AND FUTURE WORKS

The cloud infrastructure enables to support Internet scale complex applications (e.g., hospital management systems and grid systems). Complex infrastructure services and organizations will never outsource their critical applications to a public cloud without strong conformances that their needs will be enforced. This is a big challenging issue to address. A key point for solving such an issue is providing a trusted cloud scheduler which is supported by enabling the cloud scheduler to take the correct decision. Hence, such a trusted source of data is linked to both cloud user requirements and overall infrastructure properties, both are enormous, and assuring their trusted level which is our long term goal. This paper covers the most important property which is about estimating the trust status of the overall cloud infrastructure, and hence enabling cloud users to illustrate the minimal sufficient level of trust. We presented our new prototype that covers both the proposed cloud scheduler and, also our previous work which focuses on cloud trust dimensions. The main advantage of the presented prototype is that it considers complex factors that have not been considered in commercial cloud schedulers such as: taking on the entire cloud infrastructure and computing nodes' trusted level. In addition, the presented prototype enables cloud users to determine their expected trust level at the key physical resources which is hosting their virtual resources. The paper also prototype the cloud client agents that implement the cloud scheduler decisions across the overall cloud infrastructure. This paper provides a core component of our long term goal of building clouds' trusted self-managed services. Different areas of research to extend this work are identified throughout the paper, which we are planning to work on in the distant future.

REFERENCES

[1] Brian Hay, Kara Nance & Matt Bishop, (2011), 'Storm Clouds Rising: Security Challenges For IaaS Cloud Computing', Proceedings of the Hawaii International Conference on System Sciences.
[2] Frank Doelitzscher, Christian Fischer, Denis Moskal, Christoph Reich, Martin Knahl and Nathan Clarke. (2012) 'Validating Cloud Infrastructure Changes By Cloud Audits', IEEE Eighth World Congress on Services.
[3] Imad M. Abbadi & Cornelius Namiluko, (2011) 'Dynamics of trust in Clouds — Challenges and research agenda', 6th International Conference on Internet Technology & Secured Transactions.
[4] Imad M. Abbadi, (2011) 'Operational Trust in Clouds' Environment'**:** IEEE Symposium on Computers and Communications (ISCC).
[5] Jemal Abawajy, (2009) 'Determining Service Trustworthiness in Intercloud Computing Environments,' 10th International symposium on Pervasive Systems, Algorithms & Networks.
[6] Sheikh Mahbub Habib, Sebastian Ries, Max Muhlhauser, (2011) 'Towards a Trust Management System for Cloud Computing,' Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE 10th International Conference on.
[7] Sheng Di and Cho-Li Wang, (2013) 'Error Tolerant Resource Allocation and Payment Minimization for Cloud System', Ieee Transactions On Parallel And Distributed Systems, Vol. 24, No. 6.
[8] Soren Bleikertz, Matthias Schunter, Christian W. Probst, Dimitrios Pendarakis & Konrad Eriksson, (2010) 'Security Audits of Multi-tier Virtual Infrastructures in Public Infrastructure Clouds,' CCSW '10 Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop Pages 93-102.

.