International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 2, Number 3 (2012), pp. 167-171
© International Research Publications House
http://www. ripublication.com

A Naïve Approach for Managing Virtualized Resources and Optimistic cost

T.L. Surekha, Ch. Srividya, P. Ramadevi, D. Madhavi

Lecturers, Dept of I.T., V.R. Siddhartha Engineering College,
Kanuru,Vijayawada–7, A.P., India

Abstract:

Cloud computing has drawn much attention in recent years. One of its delivery models, called infrastructure as a service (IaaS), provides users with infrastructure services such as computation and data storage, heavily dependent upon virtualization techniques that offers benefits such as elasticity and cost efficiency.

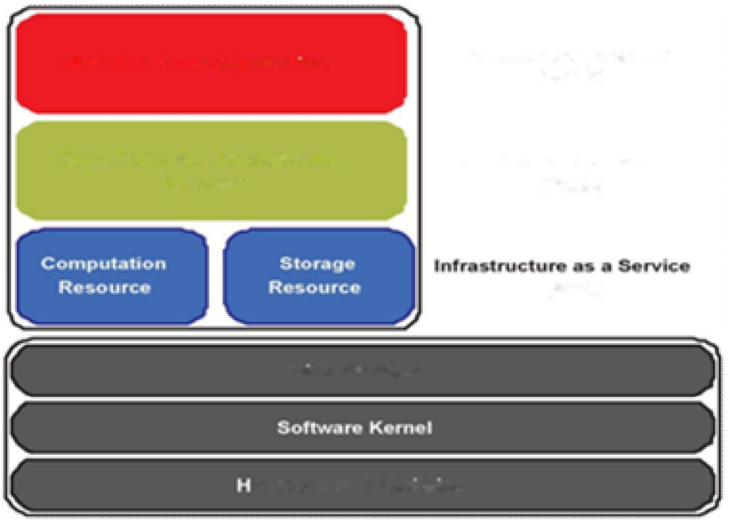
The cloud computing service providers ultimate goal is to provide maximum profits by offering cloud computing resources. We built system cost function for this queueing model. Based on the queuing model and system cost function, considering the goals of both the Cloud Computing service users and providers, we gave the corresponding strategy and algorithm to get the approximate optimistic value of service for each job in the corresponding no-preemptive priority queuing model.

Most of current IaaS service providers have adopted a user-based service model, where users are directly mapped to virtualized resources that they want to use and they are charged based on usage. Hence, user and resource management are centralized and easily administered at the IaaS provider. However, this also results in the lack of support for scalable management of users and resources, organization-level security policy, let alone flexible pricing model. Considering the increasing popularity of cloud computing, there is a strong need for a more scalable and flexible IaaS model, along with a more fine grained access control mechanism. In this paper we propose a model for provisioning and managing users and virtualized resources in IaaS to address this issue. Specifically, an additional layer is introduced to the user-based service model, and the additional layer facilitates the de-centralization of user and virtualized resource management in IaaS.

The basic idea behind role-based access control (RBAC) is to use the intermediary concept called role to provide an indirection mechanism between users and permissions. This indirection mechanism helps reduce errors in

user/permission management, support advanced features such as constraints and role hierarchy, and allow for convenient user/permission management schemes such as role-based administration and delegation. It has been also used and implemented to support the decentralization of access control management.

Background and Related work:



The general characteristics of cloud computing include on demand service, ubiquitous access, location independence, rapid elasticity, and measured service . To support these characteristics, cloud computing generally consists of three foundational components and three optional, applied components depending on its deployment models. The three foundational components are essentially those that are needed to build a collection of physical/virtualized, distributed servers for providing cloud services. They are 1) hardware and facilities,2) software kernel, and 3) virtualization, as shown in the lower part of Figure 1. The three applied components characterize and classify the services and applications of cloud computing. They are 1) computation and storage resource, 2) cloud software development platform, and 3) cloud software application. The computation and storage resource component concerns the deployment model called Infrastructure as a Service (IaaS), the cloud software development platform component pertains to another deployment model called Platform as a Service (PaaS), and the cloud software application component is related to the deployment model called Software as a Service (SaaS). Their differences are as follows; first, in the SaaS model, the cloud consumer can use the cloud provider's applications running on a cloud infrastructure which are accessible through a client interface such as a web browser. Typical examples of this type are Google Docs and Salesforce applications; in the PaaS model, the consumer can deploy onto the cloud infrastructure consumer created applications using programming languages and tools supported by the provider. Some of the examples of this type include Google App

Engine and Windows Azure and,lastly in the IaaS model, the consumer provisions processing, storage, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software. Amazon EC2 & S3 and Eucalyptus are the examples of this type. The successful implementation of cloud infrastructure requires that both foundational and applied components work together seamlessly.

Role-based security policy has attracted considerable attention in computer security communities over the last two decades, and it has grown to be a proven solution for managing access control in a simple, flexible, and convenient manner. The basic idea behind role-based access control (RBAC) is to use the intermediary concept called role to provide an indirection mechanism between users and permissions. This indirection mechanism helps reduce errors in user/permission management, support advanced features such as constraints and role hierarchy, and allow for convenient user/permission management schemes such as role-based administration and delegation . RBAC has been successfully implemented in many commercial systems including different flavors of operating systems, database systems, enterprise- based web applications. It has been also used and implemented to support the decentralization of access control management.

OUR APPROACH:

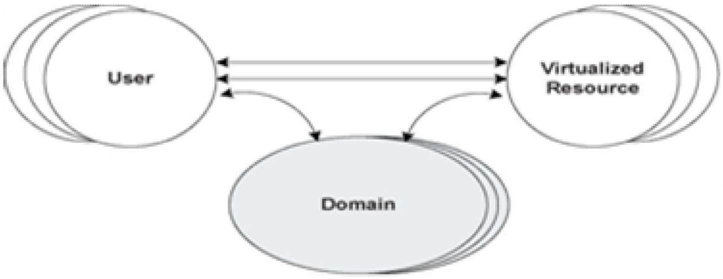
In order to support a scalable, decentralized, policy-driven scheme for IaaS, we propose a additional layer for managing users and virtualized resource in this section. First, we present the formal definitions of IaaS components along with the introduction to the notion of domain. Then redefinitions of some of role-based policy constructs follow. laaS Components:

The component of IaaS that we are most interested in for our approach is virtualized resources such as virtual machine (VM) types based on different configurations, operating system images, ramdisk images, and networking capabilities including elastic IP addresses. These kinds of virtualized resources can be found very commonly in existing IaaS platforms, with the slightly varying degree of their abstraction.

Definition 1: Let vn = {vrl' ..., vrl} denote a set of virtualized resources. A virtualized resource is represented by n-tuple, where n denotes the number of different kinds of virtualized resources serviced in the IaaS platform

Definition 2: Let U = {Ul' ..., un} denote a set of cloud users that can be uniquely identified. uvn = u x vn represents the relation of cloud user-to-virtualized resource, and the function fuv R: U -+ 2 VR maps a cloud user to a set of virtual resource

Definition 3: Let D = {d1, ..., di} denote a set of RBAC domains. UD = U x D represents the relation of cloud user to- domain, and the cloud user may or may not be associated with a domain. Similarly, DVR = D x VR represents the relation of domain-to-virtualized resource, and the virtualized resource may or may not be assigned to a domain. Lastly, UVRv represents the ternary relation of cloud userdomainvirtual resource, and the function fuv RD: U x D --- > 2 VR maps a pair of user and associated domain to a set of virtual resource.



We introduce an additional layer which performs domain-based administrative delegation, security, and user & resource management and most importantly *scheduling.*

The jobs come to the server with a Poisson distribution at a certain rate, while the process time to each job by the server is in accord with a general distribution.The algorithm is illustrated below

Assign each class in different priority with service

$\mu^0_i = t^2_i / 2t_i$ *until* $T_i < T_i^{tolerant}$; where μ^0_i stands for the minimum Cloud resources that assign to class *i* .

This procedure will guarantee the QoS for the jobs in each class.

(2) For a given determined Cloud Computing environment, according to survey from the Cloud Computing service provider, set the value of Ψ_i, δ_i for each class with different priority; and give a very small convergence value Є

3)calculate the value of $\acute{\omega}_s$ by equation

The cost function for the Cloud is:

$$\varpi(\mu) = \sum_{i=1}^{n} \omega_i(\mu_i)$$

4.) Increase each μ_i (assign more Cloud resources to class *i*), and get the value of cost function

5.)Calculate the value of

$$\Delta = |\varpi_s - \varpi_\varepsilon|$$

Conclusion:

In this paper we have discussed a novel approach to managing virtualized resources in cloud computing by introducing the notion of domain and injecting a role-based security policy support into the IaaS service model. Specifically, our approach adds an additional layer of domain to the current user-resource direct mapping mapping, and the cloud service provider delegates its administrative functions to each domain, and the domain administrator further manages users and allocated virtualized resources.

References:

[1] [1]E. S. Barka and R. S. Sandhu, "Framework for role-based delegation models," in Proceedings of i6th Annual Computer Security Application Conference, New Orleans, LA, December 2000.

[2] [2] N. Dimmock, A. Belokosztolszki, D. Eyers, J. Bacon, and K. Moody,

[3] "Using trust and risk in role-based access control policies," in Proceedings of 9th ACM Symposium on Access Control Models and Technologies, Yorktown, NY, June 2004.

[4] [3] ITU, iTU-T Recommendation X 509. information Technology: Open Systems interconnection - T he DirectolY: Public-Key And Attribute Certificate Frameworks, 2000, iSO/lEC 9594-8.

[5] [4] D. Shin, G.-J. Ahn, and S. Cho, "Role-based EAM using x.509 attribute certificate," in Proceedings of Sixteenth Annual iFfP WG il.3 Working Conference on Data and Application Security, Cambridge, UK, July 29-31 2002.

[6] [5] RBAC support for Nebula, http://nebula.nasa.govlblog/2010/jun/nebulasimplementation-of-role-based-access-control.