IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 4, Aug-Sept, 2013
ISSN: 2320 - 8791
www.ijreat.org

Calculation of Cost Factor Using Cloud Resource Allocation

Sharavana Kumar [1]

[1] Computer Science, Engineering and Technology, Bharathidasan University, Trichirappalli, Tamilnadu 620023, India

Abstract

Virtualization in cloud computing is one of the important features to share the resources among several users. So obviously the process of sharing comes under the concept of virtualization. The main objective of it is to provide shielding the differences among hardware architecture and resource utilization and flexibility. In this paper, we propose an algorithm to carry out the resource allocation process among multiple LVDs using Virtual Machine (VM). Proposed algorithm may take the parameters of hardware components such as Central Processing Unit (CPU), memory, bandwidth, processor and operating system, application being the software components. One can calculated the cost factor based on the resource utilization of the client.

Keywords: *Cloud Computing*, Virtual Machine, Logical Virtual Domain, Centeral Processing.

1. Introduction

Cloud Computing [1] has the latent to eliminate the requirements for installing high cost computing infrastructure for the users. It promises to provide a flexible process that is not limited, reachable through internet for any kind of devices. Hence devices become capable of communication between networks. In a cloud computing environment, the entire data reside over a set of interlink resources, which enables it to be accessed through VM every application runs on the VM that resides different virtualized service nodes. Basically we call it as LVD (Logical Virtual Domain) [2]. Since these data centers may be at any corner of the world beyond the reach and control of users, there are some privacy challenges that need to be understood and taken care of.

Internet has been emerging trend towards the various technologies that have been developed. Cloud computing facilitates non existence of hardware resources and maintenance cost. Hence accessibility hassles around the globe and software up-gradation problems may be overcome.

There are three types of services offered by the cloud computing. There are Infrastructure as a Service (IaaS), Plat form as a Service (PaaS), Software as a Service (SaaS) [3].

IaaS is the lowest among all the layers, it provides basic infrastructure support, PaaS is the middle layer, it provides the platform oriented services, SaaS is the top layer among all the layers it provides a complete application offered as service on demand. It eliminates the need to install and run and maintenances of the application on the customer's local computer.

Cloud computing is user friendly model, the way which provides the process based on pay-per-use service. Cloud enables the way to add capabilities without the need of brand new infrastructure, new personnel. The owners of cloud application should provide the security polices to control the access for those who request resources. Moreover, the owner should also provide properly configured software and hardware, enforcement of the access control policy on resource allocation [4] [5] which are from multiple service nodes. In Information Technology infrastructure could require act of useful for number of services, they lead to wrong configuration. To address the above problems we introduce the Multilayer Access Control for LVD.

A VM is a software usually resides on a machine in order to utilize the same hardware for different operating systems on a single machine. It may also be used to support a real time operating system at the same time as a high-level operating system and they can allow you to run multiple operating systems simultaneously on a single machine. Virtualization has its own dividing, which cause a single physical server into multiple logical servers. The hypervisor is the most basic virtualization component. It is the software that disengages the operating system and applications from their physical resources.

.

2. Related Work

The works reported so far related to this work mostly focus on isolation control and isolation management, but cannot be applied to LVD environment with more security requirements, such as information flow control.

sHype[6] enables the MAC frame work integrated to Xen hypervison and implementation of xen security module , it also encompasses the sharing resources and isolation , moreover the security level is good by using Simple Type Enforcement (STE) , China Wall (CW), Since cloud is distributed the MAC is not applicable for this single LVD.

Shamon [7] is based on sHype and labeled IPsec, it implements MAC enforcement in two layers namely Xen virtual machine monitor (VMM), SELinux and IPsec network, it fails to covert channel control because of no specific MAC policy.

In Trusted Virtual Datacenter (TVD) [8] based on trusted virtual domain and integrates virtualized based security and system management, it enables the containment guarantee for Datacenter and Trust Management, inter and intra TVD.

Resource allocation is an essential and growing part of many data center management problem such as virtual machine placement in data center, network virtualization, In VM allocation strategy the constraints that involve individual server capacity, network bandwidth capacity in the data center.

Starting of VMs has to be in a secure manner, in order to avoid fault resource sharing due to wrong configurations of VM, The requested client should get the resource as they want with all constraints , the problem is that virtual machine allocation and band width allocation strategy.

The proposed system supports access control on the LVD's in cloud environment, it also provides the policy management mechanism. It also has two nodes management node and service node. These two nodes take care of virtual machines starting in a secure manner, and there are some specified policy has applied to control the band width.

3. Process Flow

The process of the formalizing LVD modules will be first begin with Agent factory, the Agent factory will request the master to deploy VM, if the request granted from master then its response to agent factory, the self test will be there to ensure the trust measurement, then the agent will be create by the agent factory, if its success means then VM will create with secure module with local policy.
.

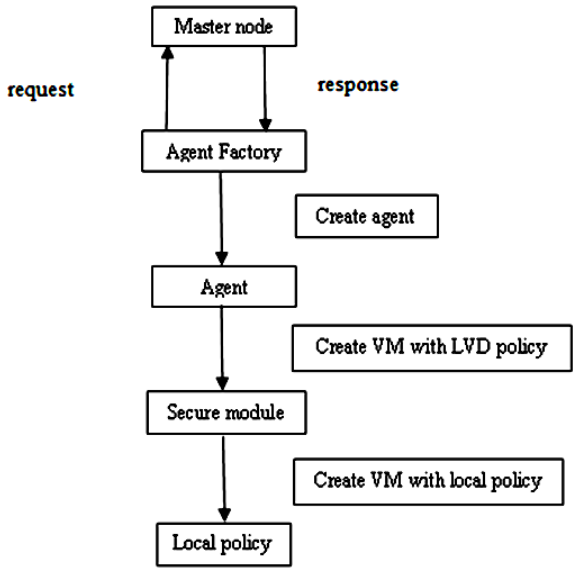

Fig: 1 Process Flow of creating a VM

4. Starting a VM

Starting a VM on a node may cause a conflict, because it attached access control label may conflict with the labels of the VMs that have already been started on the same node. The following algorithm overcomes the problem. The following algorithm in fig 2 shows the starting a VM as a single service node. The access control on a service node and the access control for VM joining an LVD.

```
If local node = empty
VM is permitted to start
Put VM label T1 in local node
Else if (T1 is in local node)
VM is permitted to start
Else if T1 is not in clash with any VM label listed
VM is permitted to start
Put the VM label in local node
Else
VM start request is denied
```

Fig: 2 Algorithm for starting a VM

5. Algorithm to keep track the resource for ultimate utilization

The first step is to verify all the VM that are resides in LVDs, then gather the information about VM , which is not in use and also get the currently running VM. Once it has been done, it is possible to identify the VM that are free. Then define the Deployable to NULL, check the size of the used VM list is equals to available VM list then clear the VM list to zero then iterate the VM with respect to all the LVD, and then check the VM not in used VM list then, if the condition is true then add that VM to used VM list, and then set the deploy VM to VM. Else return the deploy VM. With some enhancement of the proposed system the process may lead to self adaptable cloud [9], that can aware of all those resource allocation by their own entities.

```
Verify All LVD
 Get the available VM list
Get the used VM list
Deployable VM =null
If
Size(used VMList)=size(available VMList) then
Clear used VMList
End if
For VM in (LVD(i)) do
If VM not in used VMlist then
Add VM to used VMList
Deployable VM=VM
Break
End if
End for
Return deployable VM
```

Fig: 3 Algorithm for resource utilization

Where VM = virtual machine, VM List= available virtual machine, LVD=Logical Virtual Domain.

6. Simulation Results:

The implementation of the project is performed with java language. There are plenty of pre-defined packages to implement the cloud concept. The platform is very useful to finish the process under one roof.

Once the boundary has been specified among the entire available center, then the client geographical area has to be chosen consequently .The client configuration is to be specified according to the client's requirements. Finally the overall response time and data center processing time has calculated according the performance of the VM processing based on the user request.

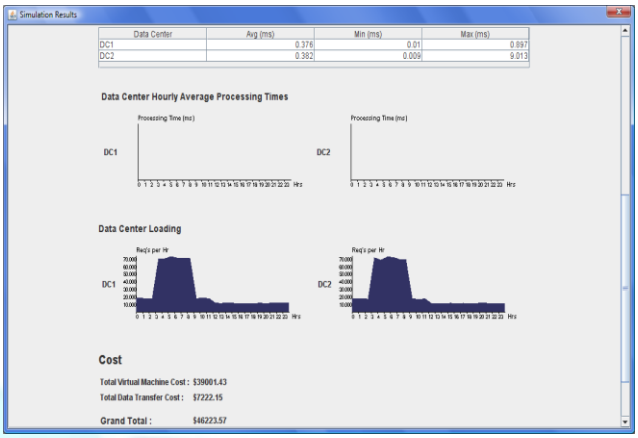

Fig: 4 Output Cost Calculations

The overall cost has been calculated after the client's process completion according to the usage of the VM, amount of time resource used by the client. VM cost and data transfer cost has also been calculated with respect to the data center, and then the sum of those cost factor have been calculated in order to get the grand total. That is the total cost for the client.

7. Conclusion

By using resource allocation algorithm the client benefits by the resource which is allocated to him. According to the usage of the allocated resource of the client, the client is requested to pay the amount of his usage. A specific algorithm is used to start the VM in order to avoid the unauthorized usage of the VM other than the clients. Hence it enables the restriction of access control by providing the constraint, via which constraints that help can be authorized. The client can attain the resource easily whenever the requested resource is not available at the specific node by resource allocation algorithm. Finally the cost of utilization is compared according to the resource requested based on the configuration which the client have been used. Therefore we have achieved the goals of access control and resource-sharing control among multiple VMs.

References

[1] Foster, I., Zhao, Y., Raicu, I., Lu, S.: 'Cloud computing and grid computing 360-degree compared'. Proc. Grid Computing Environments Workshop (GCE'08), pp. 1–10.
[2] Weizhong Qiang1, Deqing Zou1, Shenglan Wang1, Laurence Tianruo Yang2, Hai Jin1, Lei Shi1 : 'Cloud Ac: Cloud Oriented Multilayer Access Control System for Logical Virtual Domain'. 7, Iss. 1, pp. 51–59.
[3] Bhadauria, R., Chaki, R., Chaki, N., & Sanyal, S. (2011). A survey on security issues in cloud computing. arXiv preprint arXiv:1109.5388.

[4] Pawar, C. S., & Wagh, R. B. (2012, December). Priority Based Dynamic resource allocation in Cloud Computing. In Cloud and Services Computing (ISCOS), 2012 International Symposium on (pp. 1-6). IEEE.
[5] Patel, R., & Patel, S. (2013). Survey on Resource Allocation Strategies in Cloud Computing. International Journal of Engineering, 2(2).
[6] Sailer, R., Jaeger, T., Valdez, E., et al.: 'Building a MAC-based security architecture for the Xen open source hypervisor'. Proc. 21st Annual Computer Security Applications Conf. (ACSAC' 2005), Miami, FL, USA, December 2005, pp. 276–285.
[7] McCune, J., Berger, S., Cacerres, R., Jaeger, T., Sailer, R.: 'Shamon: a system for distributed mandatory access control'. Proc. 22st Annual Computer Security Applications Conf., Miami, FL, USA, December 2006, pp. 23–32.
[8] Berger, S., Caceres, R., Pendarakis, D., et al.: 'TVDc: managing security in the trusted virtual datacenter', ACM SIGOPS Oper. Syst. Rev., 2008,42, (1), pp. 40–47M.
[9] Maurer, M., Brandic, I., Emeakaroha, V. C., & Dustdar, S. (2010, July). Towards knowledge management in self-adaptable clouds. In Services (SERVICES-1), 2010 6th World Congress on (pp. 527-534). IEEE.
[10] http://www.cloudfoundry.com
[11] http://www.CIO.com
[12] http://www.vmware.com