

MANAGEMENT OF VIRTUAL MACHINE AS AN ENERGY CONSERVATION IN PRIVATE CLOUD COMPUTING SYSTEM

Akhmad Fauzi¹, Edy Mulyadi², Abdullah Fadil³, Mohammad Idhom⁴

UPN "Veteran" Jatim

Email: akhmadfauziupn@gmail.com¹, edlmulyadi@gmail.com²,
ziyanza@yahoo.com³, moh.idhom@gmail.com⁴

ABSTRACT

Cloud computing is a service model that is packaged in a base computing resources that can be accessed through the Internet on demand and placed in the data center. Data center architecture in cloud computing environments are heterogeneous and distributed, composed of a cluster of network servers with different capacity computing resources in different physical servers. The problems on the demand and availability of cloud services can be solved by fluctuating data center cloud through abstraction with virtualization technology. Virtual machine (VM) is a representation of the availability of computing resources that can be dynamically allocated and reallocated on demand. In this study the consolidation of VM as energy conservation in Private Cloud Computing Systems with the target of process optimization selection policy and migration of the VM on the procedure consolidation. VM environment cloud data center to consider hosting a type of service a particular application at the instance VM requires a different level of computing resources. The results of the use of computing resources on a VM that is not balanced in physical servers can be reduced by using a live VM migration to achieve workload balancing. A practical approach used in developing OpenStack-based cloud computing environment by integrating Cloud VM and VM Placement selection procedure using OpenStack Neat VM consolidation. Following the value of CPU Time used as a fill to get the average value in MHz CPU utilization within a specific time period. The average value of a VM's CPU utilization in getting from the current CPU_time reduced by CPU_time from the previous data retrieval multiplied by the maximum frequency of the CPU. The calculation result is divided by the making time CPU_time when it is reduced to the previous taking time CPU_time multiplied by milliseconds.

Keywords: cloud computing, virtual machine, energy conservation.

INTRODUCTION

Cloud computing is a service model that provides convenience to the end-users and brokers. These services in the pack in the base of computing resources that can be accessed through the Internet on demand. These resources are presented to the customer self-service through the efforts of management and interaction with minimal service providers (Mell & Grance, 2011). Cloud Service Provider puts the computing resources in the data center.

In terms of management, data center architecture in cloud computing environments are heterogeneous and distributed, composed of a cluster of network servers with different capacity computing resources in different physical server. Cloud computing service providers still continue to look for a proper solution in order to optimize the cloud data center infrastructure. (Zhang, et al., 2010). The challenge is that if the capacity of the computing resources provided by the data center cloud more than the demand then the service provider would suffer losses due to the overhead of operating costs and processes occurring in the data center will experience a shortage utilization (under utilization). Conversely, if the capacity of computing resources provided by the data center experienced a shortfall due to popular demand, the service provider will lose revenue as a result in the stay by the customer and the processes that occur in the cloud data center will have excess usage (over utilization) (Armbrust, et al., 2009).

To overcome the problem of demand and the availability of the cloud services which are fluctuating, then the cloud data center must be made elastic. The technology that makes it possible to realize this is virtualization. Virtual Machine (VM) is a representation

of the availability of computing resources that can be dynamically allocated according to demand.

VM consolidation strategy can be done by considering when, where and where VM must migrate. The strategy is broken down into sub decisions namely overloading host detection, underloading host detection, and VM and VM placement selection (Beloglazov & Buyya, 2012). The host overloading detection and the host underloading detection are used to determine when a host is considered experiencing overloaded or underloaded.

In the consolidation procedure of VM, host overloading detection is often a major focus of research in order to balance the workload in associate with the aim of saving energy consumption in the data center cloud (Masoumzadeh & Hlavacs, 2013).

VM migration decision is based on a defined policy in the capture of resource utilization and server computing in a deal offense defined through a Service Level Agreement (SLA).

Cloud computing environments that require dynamic and distributed decision-making procedures and strategies within the VM consolidation will be made dynamically as well. So in this study the proposed electoral policy optimization and placement of VM on VM consolidation procedures in cloud data center environments by considering hosting certain types of service applications on a VM instance that require various computing resources levels. Accumulated use of computing resources on a VM that is not balanced in physical servers can be reduced by using a live VM migration to achieve workload balancing. The practical approach will be used in developing OpenStack-based

cloud computing environment by integrating Cloud VM and VM Placement selection procedure using OpenStack Neat VM consolidation (Beloglazov & Buyya, 2012).

The term cloud computing or cloud computing was inspired by the image of cloud (cloud) that are commonly used to describe the current Internet network diagram. In fact there is a link between cloud computing and the Internet. Cloud computing is a client-server model, where resources (resources) such as servers, storage, network and software can be viewed as a service that can be accessed by users remotely and at any time. Users can enjoy a wide range of services provided by the provider of cloud computing without the need for too much ask for technical assistance or support from the provider. Cloud computing service model can be illustrated by (Mell & Grance, 2011).

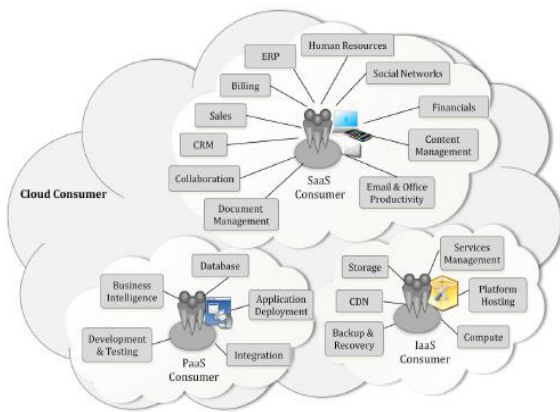


Figure-1. Model of cloud computing services (Zhang, et al., 2010)

Cloud computing infrastructure such as servers, storage, network and various software known as cloud and placed in the data center.

Virtual Machine Migration

The migration process demonstrate the functionality that removes the process from one computer to another. However, there are differences in the operating concept of virtual machine migration and migration processes (Khyaita, et al., 2012). Migration of the entire OS and all applications as one unit to avoid the many difficulties faced by the process approach the level of migration. Narrow interface between the virtual OS and the virtual machine monitor (VMM) to avoid residual dependency issues. Migration at the level of the entire virtual machine in the state means that the memory can be transferred consistently and efficiently. Migration of virtual machines allows separation of interests between users and data center operators. Virtual Machine (VM) migration can be used for consolidation VM dynamically based on analysis of historical data on the use of resources by the VM. The problem is divided into four parts: (1) determine when the host is considered underloaded which leads to the need to migrate all of the VM from the host and switch hosts into sleep mode, (2) determine when the host is considered overloaded leading to the need of migration from one or more VM from the host to reduce the burden, (3) selecting the VM to be migrated from host overloaded, and (4) find a new

placement of the selected VM to migrate from host overloaded (Beloglazov & Buyaa, 2013).

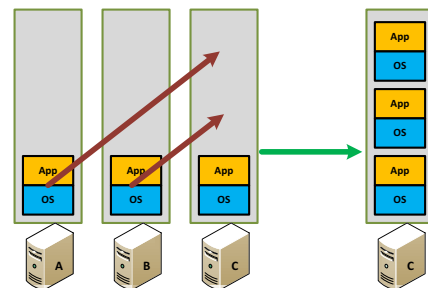


Figure-2. Virtual machine migration to consolidation

Cloud Computing-based OpenStack CloudOpenStack project is a cloud computing platform based on open source that is aimed at public and private cloud. The project initially focused on the presentation of Infrastructure as a Service (IaaS). The main objective of the project is to produce a ubiquitous OpenStack Open Source Cloud Computing platform that will meet the needs of public and private cloud regardless of size, with simple implementation and mass meet scalability.

Openness is a core value behind the project with open standards and open source. OpenStack has been released under the Apache 2.0 license. OpenStack also supports standard through the OpenStack API. Rackspace and NASA originally OpenStack project has commenced in September 2012. Companies such as AT & T, Rackspace, HP, IBM, Canonical, Red Hat, SUSE, Cisco, Dell, and Intel are examples of the 1,505 companies that support and the community very active in contributing to developing OpenStack.

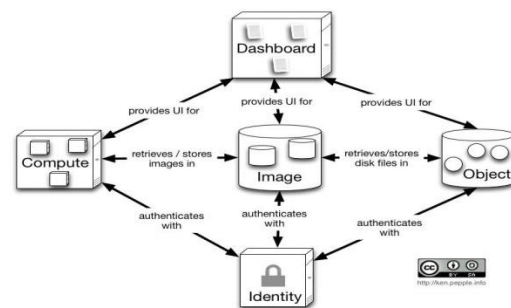


Figure-3: Architecture OpenStack Cloud (Khyaita, et al., 2012)

OpenStack Neat is a project intended to provide an extension to the OpenStack Cloud implementing dynamic consolidation of Virtual Machines (VM) using live migration. The main objective is the dynamic VM consolidation to improve the utilization of physical resources and reduce energy consumption to reallocate the VM using live migration of resources according to demand real-time and divert hosts idle into sleep mode.

For example, it is assumed that the two VM placed on two different hosts, but the capacity of the combined resources needed by the VM to serve the load current can be supplied by one of the hosts. Then, one of the VM can be migrated to a host of support services other VM on the destination host, and the host that is idle can be diverted into a low power mode to save energy. Neat OpenStack consists of a number of components and data storage

units, some of which are used to compute hosts, and some on the Management of hosts

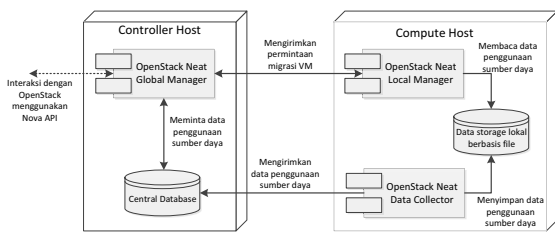


Figure-4. Framework OpenStack Neat (Beloglazov, 2014)

Design of system implementation

A series of tests will be done in the cloud-based data center using the Framework OpenStack OpenStack Cloud Neat. Environmental tests are used as in Figure 5.

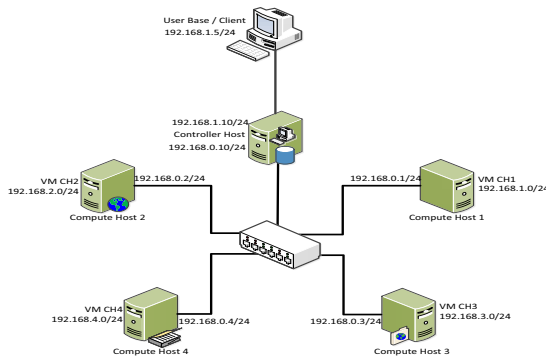


Figure-5. Testbed Network Topology (Beloglazov & Buyya, 2013)

RESEARCH METHODOLOGY

This research was conducted in testbed environments or testing in real terms in cloud computing systems by integrating a dynamic procedure VM consolidation. Stages of implementation of the research method used is as follows:

- 1) Development of testing in real environment by implementing and integrating OpenStack cloud platform OpenStack framework -Neat as VM dynamic consolidation procedures. At this stage the environment resulting cloud computing systems that can perform the function VM migration between the originating host to a destination host through a dynamic mechanism VM consolidation.
- 2) Design and implementation of procedures VM consolidation dynamic environment system testing OpenStack-based cloud computing cloud with the methods that have been proposed. At this stage would result VM module selection, placement VM, host overload detection and underload detection host.
- 3) The design of test scenarios that will be used to test the performance of the system. Designed test scenarios include variations based resource intensive workloads that include CPU-intensive, Memory-Intensive, DiskIO-intensive and Network-Intensive. At this stage that is used to generate scenarios for system testing process.

- 4) Testing of the entire system. At this stage of testing for the selection mechanism VM that have resulted from the second phase and test scenarios resulting from the third stage.

The development of the election program and the migration of VM through a procedure of consolidation VM dynamically performed by modifying the modules available in the framework OpenStack-Neat (data modules collector, local manager, underload, overload, VM selection, global manager and VM placement as well as modules other support).

RESULT AND DISCUSSION

Models of cloud computing virtualization technology implemented in cloud computing systems development model based private cloud that will build and manage cloud infrastructure independently using the open source platform OpenStack-based cloud. The OpenStack cloud can be implemented on multiple compute nodes to build distributed real testbed environment.

Host or Node used in testbed environments will use CentOS-based Linux distribution. CentOS installation stage following the standard process is described in detail in the Red Hat Enterprise Linux Installation Guide (). HDD Partition Scheme In Host OpenStack Cloud

To support the needs VM live migration, the distributed file system (DFS) must be provided in order to share storage are also replicated between multiple servers via ethernet. The implementation will be supported by providing a different disk partition on the host used in OpenStack cloud. Disk partitioning scheme results in compute host consists of standar partition for the operating system lv_root, lv_home and lv_swap are grouped in vg_base partition. Special partition lv_gluster prepared for distributed file system (DFS) based GlusterFS contained in vg_gluster group.

Network Gateway OpenStack Cloud

Internet access is required by the host OpenStack cloud in order to facilitate the installation process and a package of modules required to form a testbed environment. One compute hosts (compute1) was Chosen as the gateway to the IP address configuration as in the table () to enable the network address translation (NAT) using iptables as in the following description:

```
#iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
#iptables -A FORWARD -i eth1 -j ACCEPT
#iptables -A FORWARD -o eth1 -j ACCEPT
```

So that data packets can be forwarded from another host to the internet via compute1, then IP forwarding facility can be activated with the command:

```
sed -i 's/net.ipv4.ip_forward = 0/net.ipv4.ip_forward = 1/g' /etc/sysctl.conf
```

Using DFS implementation GlusterFS

The main package and supporting GlusterFS taken from the official website with the command:

```
#yum -y install wget \
http://download.gluster.org/pub/gluster/glusterfs/L
ATEST/CentOS/glusterfs-3.3.0-1.el6.x86_64.rpm \
http://download.gluster.org/pub/gluster/glusterfs/L
ATEST/CentOS/glusterfs-server-3.3.0-1.el6.x86_64.rpm \
```

In the host controller in the host to do initialization is involved in the formation of DFS to perform the command:

```
#gluster peer probe compute1
#gluster peer probe compute2
#gluster peer probe compute3
#gluster peer probe compute4
```

Volume gluster for replication between four gluster host created with the command:

```
#gluster volume create VM-instances replica 4 \
compute1:/export/gluster
compute2:/export/gluster \
compute3:/export/gluster
compute4:/export/gluster
```

Results of the implementation of DFS using GlusterFS is :

```
Volume Name: VM-instances
Type: Replicate
Volume ID: 7b0e3467-33bd-42df-962d-
288d567a7a11
Status: Started
Number of Bricks: 1 x 4 = 4
Transport-type: tcp
Bricks:
Brick1: compute1:/export/gluster
Brick2: compute2:/export/gluster
Brick3: compute3:/export/gluster
Brick4: compute4:/export/gluster
```

Implementation Using KVM Hypervisor

Virtualization run only on compute hosts using KVM. KVM host controller does not need because it is not running VM instance. To enable KVM on compute hosts, it must be ensured that the machine supports Intel VT or AMD-V for virtualization in hardware. The feature can be activated via the basic input output system (BIOS). Implementation KVM on CentOS Linux can be done via the command:

```
yum -y install kvm qemu-kvm qemu-kvm-tools
```

Abstraction and Hypervisor API Implementation using libvirt Communication between the KVM with OpenStack Cloud bridged by a user application programming interface (API) using libvirt. OpenStack using libvirt as it supports a wide range of hypervisors including KVM and Xen. Libvirt package can be installed using the command:

```
#yum -y install libvirt libvirt-python python-
virtinst avahi dmidecode
```

To support live migration capabilities on OpenStack Cloud VM, then libvirt configuration needs to be modified to enable communication via TCP without authentication.

```
#sed -i 's/#listen_tls = 0/listen_tls = 0/g'
/etc/libvirt/libvirtd.conf
#sed -i 's/#listen_tcp = 1/listen_tcp = 1/g'
/etc/libvirt/libvirtd.conf
#sed -i 's/#auth_tcp = "sas1"/auth_tcp = "none"/g'
/etc/libvirt/libvirtd.conf
#sed -i 's/#LIBVIRT_ARGS="--
listen"/LIBVIRT_ARGS="--listen"/g'
/etc/sysconfig/libvirtd
```

Implementation of cloud computing using OpenStack Platform Node or host that is used for the implementation of cloud computing using the OpenStack platform is divided into the controller and compute hosts. On each host needed an update repository to retrieve the packages needed to build OpenStack Cloud with the command:

```
#yum install -y
http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-
6-8.noarch.rpm
```

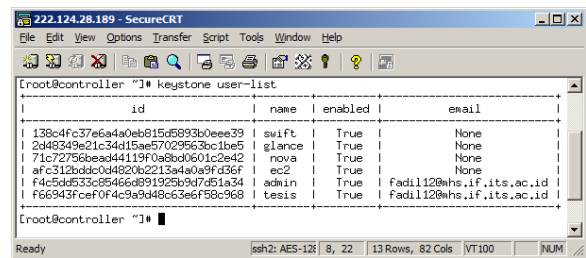


Figure-6. At the OpenStack User List

Glance component package is a VM image management service that is used in OpenStack Cloud. Glance requires MySQL to store the VM image metadata. To check service Glance running properly can be done by downloading Ubuntu Cloud Image and put it into the command Glance:

```
#wget http://uec-
images.ubuntu.com/precise/current/precise-server-
cloudimg-amd64-disk1.img
#glance add name="ubuntu" is_public=true
disk_format=qcow2 container_format=bare <
precise-server-cloudimg-amd64-disk1.img
```

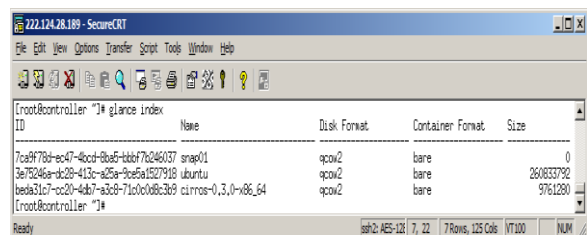


Figure-7. List VM image on OpenStack Cloud

Integration Framework Neat on OpenStack
 OpenStack Cloud OpenStack-Neat is a module used by OpenStack Cloud in implementing the consolidation of a VM dynamically using a live VM migration to improve the performance of cloud services, maximizing the utilization of physical resources and reduce energy consumption by reallocating VM using live VM migration based on demand of resources in real terms and disable the host is not busy into power saving mode. In research conducted, can be integrated into Neat OpenStack OpenStack Cloud environment without changing the configuration of existing and already running. The installation procedure OpenStack OpenStack Cloud Neat on the environment, namely:

- 1) Duplication OpenStack neat on each host or node that is used with the command: #git clone <https://github.com/beloglazov/openstack-neat.git>
- 2) Adjust the configuration file neat.conf on environmental conditions OpenStack Cloud.
- 3) Package OpenStack Neat installed on each node of the repo directory by running the command: #python setup.py install
- 4) Run the service of the host controller with the command: #. / All-start.sh

CONCLUSIONS

Consolidation VM can dynamically react to the increasing demand for resources to reallocate the VM back when necessary to avoid degradation in the performance of cloud computing system as a whole. VM consolidation procedures that were implemented into OpenStack-Neat can be divided into several components, namely the data collector, the local manager and global manager.

The components used to handle the consolidation procedure is to decide when the host is considered as host underload and overload, the selection of VM to migrate from overload host and placement of the selected VM on the selected host. Value CPU Time used as fill to get the average value in MHz CPU utilization within a specific time period.

The average value of a VM's CPU utilization in getting from the current CPU_TIME is reduced by CPU_TIME from the previous data retrieval multiplied by the maximum frequency of the CPU. The calculation result is divided by the time making CPU_TIME when it is reduced to the time taking previous CPU_TIME multiplied by milliseconds.

REFERENCES

Armbrust, M. et al., 2009. Above the Clouds: A Berkeley View of Cloud Computing. *Electrical Engineering and Compute Sciences University of California at Berkely*.

Beloglazov, A., 2014. *OpenStack Neat*. [Online] Available at: <http://www.openstack-neat.org>

Beloglazov, A. & Buyaa, R., 2013. Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers Under Quality of Service Constraints. *IEE Transactions on parallel and distributed system*.

Beloglazov, A. & Buyya, R., 2012. Optimal Online Deterministic Algorithms and Adaptive Heuristics.

Concurrency and Computation: Practice and Expreience, pp. 1397-1420.

Beloglazov, A. & Buyya, R., 2013. *OpenStack Neat: A Framework for Dynamic and Energy-Efficient Consolidation of Virtual Machines in OpenStack*. s.l.:Cloud Computing and Distributed Systems (CLOUDS) Laboratory.

Corradi, A., Fanelli, M. & Foschini, L., 2014. VM Consolidation: A real case based on OpenStack Cloud. *Future Generation Computer Systems* 32, pp. 118-127.

Khiyaita, A., Zbakh, M., Bakkali, H. & Kettani, D. E., 2012. Load Balancing Cloud Computing:state of art. *Network Security and Systems (JNS2)*, pp. 106-109.

Masoumzadeh, S. & Hlavacs, H., 2013. s.l., IEEE, pp. 332-338.

Mell, P. & Grance, T., 2011. *The NIST Definition of Cloud Computing*, s.l.: National Institute of Standard and Technology.

OpenStack, 2013. *OpenStack Cloud Administrator Guide*, s.l.: docs.openstack.org.

Zhang, Q., Cheng, L. & Boutaba, R., 2010. Cloud Computing: state-of-the-art and Research Challenges. *Journal of internet services and applications*, pp. 7-18.