S6–02

# Virtual Machine Migration for Dynamic Server Resource Management in Cloud Computing Environments Based on OpenStack

*Kim Ryo Chol**, *Kim Sun Hui*

School of Information Science and Technology, Kim Chaek University of Technology, Pyongyang, DPRK
*Corresponding author: Email: krc771218@star–co. net. kp

## Summary

With the emergence of cloud computing, many researchers have focused on building data centers and establishing public and private clouds. While the number of data centers is growing because of the low utilization rate of resources, many servers in data centers are in a low load condition, resulting in idle resources. With this trend, there occurred several problems including more rational use of hardware resources, reduced energy consumption of server clusters in data centers, improvement in their efficiency, etc. Virtual machine (VM) migration can effectively relocate virtual resources, so it has become an important resource management method in server clusters and data centers. In this paper, we present the taxonomy to categorize server resources. And a virtual machine migration scheme is investigated for server load balancing and power reduction in cloud computing environments based on OpenStack.

*Keywords*: Cloud computing; OpenStack; Resource management; Virtual machine

## 1. Introduction

Cloud computing technology has become the solution to idle hardware resource and network server load, bringing about improvement in the ways of using computer resources. Today, there are many resources such as disk, memory, and CPU in modern computers which are based on Von Neumann Organization. The resources of a single physical machine are divided into multiple isolated virtual resources by using some virtualization software. The isolated virtual environment is called virtual machine (VM). Resources including CPU, memory, network bandwidth, and system cache are dynamically provisioned to a VM based on the performance requirements. Virtualization runs numerous VMs on a single hardware resource. Virtualization technology is the key point for deploying infrastructure services in cloud environment. It facilitates server consolidation, load balancing, and power saving [3, 5, 9].

Nowadays Eucalyptus, the commercial cloud computing system, is the software of Platform as a Service for workstations and provides an extensible and real–time cloud environment. And OpenNebula is also used for dynamic allocation of resources and provides Infrastructure as a Service environment for cloud computing facilities.

OpenStack is a collection of open source components to deliver public and private clouds that supply Infrastructure as Service [6, 8, 10].

OpenStack includes following constituents:

- OpenStack Compute("Nova"): Software to orchestrate, manage, and offer virtual machines.
- OpenStack Object Store("Swift"): Software for the redundant storage of static objects.
- OpenStack Image Service("Glance"): Provides query and storage services for virtual disk images.

This paper aims at the rational management of server resource for server load balancing and reducing energy consumption by virtual machine migration using OpenStack Compute.

## 2. Dynamic resource management using VM migration

### 2.1 Taxonomy of server resources

Underutilization of server resources leads to substantial rise in power consumption of data centers. Server consolidation is among the methods that improve server resource utilization efficiency by eliminating under–loaded servers to reduce hardware cost and power consumption within data centers.

In this paper, server resources are classified into three categories:

Physical Servers

A set of physical servers

$$R = (r_1, r_2, \cdots, r_n) \tag{1}$$

where $n$ is the number of physical servers.

Running Virtual Machines

A set of all VMs running on a physical server $r$

$$V_r = (v_1, v_2, \cdots, v_m) \tag{2}$$

where $m$ is the number of VMs on a physical server $r$.

Idle Virtual Machine

A set of all VMs that are waiting on a physical server $r$

$$IV_r = (iv_1, iv_2, \cdots, iv_k) \tag{3}$$

where $k$ is the number of waiting virtual machines on a physical server $r$.

The current running server satisfies the condition $\exists v \in V$ where $r \in R$ is a running physical server.

The current running under–loaded server satisfies the condition $\exists iv \in IV$ where $r \in R$ is a running physical server.

This under–loaded server $r$ uses a fraction of their total capacity, so the current running virtual machines are shifted from this under–loaded server to underutilized servers in order to increase resource usage efficiency, thus an idle physical server is automatically switched to a low–power mode and we can reduce the overall energy consumption.

### 2.2 Virtual machine migration based on OpenStack

Virtual machine migration can effectively relocate virtual resources and it has become an important resource management method in clusters and data centers [1]. The migration controller accesses the servers' usage statistics (CPU, memory, network, etc.) using a data collector running within physical servers.

The data collector periodically collects the CPU utilization data for each VM running on the physical servers. The CPU usage $U_i(t_0, t_1)$ of a VM $i$, which is calculated in the measuring interval $[t_0, t_i]$ as shown (4).

$$U_i(t_0, t_1) = \frac{n_i(\tau_i(t_1) - \tau_i(t_0))}{(t_1 - t_0)} \quad (4)$$

where $n_i$ is the number of virtual CPU cores allocated to VM $i$ and $\tau_i(t)$ is the CPU time consumed by VM $i$ up to time $t$.

The CPU usage by the set of VMs allocated to the physical server $r$, $C_r(t_0, t_1)$, is calculated as shown in (5).

$$C_r(t_0, t_1) = \sum_{i \in V_r} U_i(t_0, t_1) \quad (5)$$

where $V_r$ is the set of set of VM allocated to the physical server $r$.

The data collector uses the maximum amount of RAM that can be used by a VM statically, rather than the real–time RAM consumption. The reason for that is that RAM is a more critical resource compared with CPU, as an application may fail because of insufficient RAM, whereas insufficient CPU may just slow down its execution.

Based on the servers' usage statistics the problem of dynamic server resource management consists in splitting into three sub–problems: (1) server under–load/overload detection; (2) Selecting VMs to migrate; and (3) VM placement.

Server under–load/overload detection

After measuring the n latest CPU utilization of the server $r$, the mean $C_r$should be calculated using regression model as shown (6).

$$C_r = a_1 C_r^1 + a_2 C_r^2 + \cdots + a_n C_r^n \quad (6)$$

where $C_r^i$ is the $i^{th}$ CPU utilization measurement and $a_i$ is the $i^{th}$ coefficient.

The reason for giving latency is because we can avoid migration of VMs in the case of server load peak.

If the mean CPU utilization is lower than the threshold, we can detect a server under–load situation. If a server is considered to be under–loaded, all VMs should be migrated from it, and the server should be switched to a low–power mode.

The Linux OS provides an API to programmatically switch the physical server into the sleep mode. The machine can be reactivated over the network using the Wake–on–LAN technology.

If the mean of the $n$ last CPU utilization measurements is higher than the specified threshold, we can detect overload situations. Once a server overload has been detected, it is necessary to determine what VMs are the best to be migrated from the server. This problem is solved by VM selection algorithms. Selecting VMs to migrate

An example of such an algorithm is randomly selecting a VM from the set of VMs allocated to the overloaded server, but the main goal of migration is to minimize total migration time and reduce network traffic.

In general, migration of VMs needs to save all data and status in it, transfer to the target physical server through network, and load in RAM [2, 4, 7].

We propose an algorithm shown in Algorithm 1. This algorithm first selects VMs with the threshold amount of RAM to reduce the migration time. Then, out of the selected subset of VMs, the algorithm selects the VM with the maximum CPU utilization averaged over the last n measurements to maximally reduce the overall CPU utilization of the server.

**Algorithm 1.** Selecting VMs to migrate from an overloaded server

```
Input: n, vm_ary, vm_ram_ary, threshold_ram
Output: a VM to migrate
max_cpu_usg←0;
selected_vm←none;
for( each vm in vm_ary) {
  if ( vm_ram_ary[vm] < threshold_ram) {
    for( each core in vm_array) {
      mean_core ← mean of last n values
                  of core utilization;
      if (max_cpu_usg < mean_core) {
        max_cpu_usg ← mean_core;
        selected_vm ← vm;
      }
    }
  }
}
return selected_vm;
```

VM placement based on OpenStack

Nova–Compute service determines the server for a VM instance to be launched on.

VM placement includes of filtering and weighting. For each VM placement request, Nova–compute selects servers with available resources to the hardware request. A more complex filtering mode can be used. That is, if a server is selected for a VM according to its CPU requirements, the server is confirmed if it just satisfies the RAM requirements.

After selecting a couple of physical servers, each server should be weighed. The greater the cost of VM creation, the greater its weight is. We can select the physical server that has the smallest weight and provision instances on it as shown in Fig. 1.

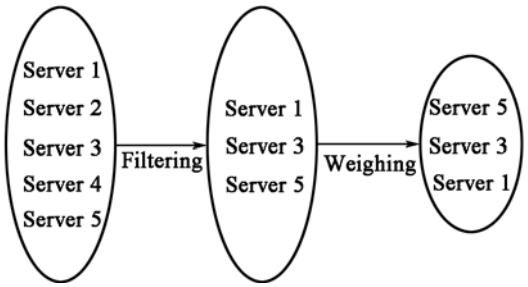


**Figure 1.** Selection of physical servers

But all VM instances are not invariant according to the use of resources and the need of computing capacity, if the server remains in an idle state for a while, it results in load imbalance.

## 3. Performance evaluation

The test bed used for performance evaluation of the system consisted of the following set of servers:

Intel(R) Xeon(R) CPU (four cores)×4, at 3.40GHz

16–GB DDR3–1333

Western Digital 1TB, 7 200RPM SATA II
Dual Gigabit Ethernet

As shown in Fig. 2 and 3, the experiment results of proposed method show that server load balancing could be obtained.


**Figure 2.** Utilization of CPU resources without VM migration

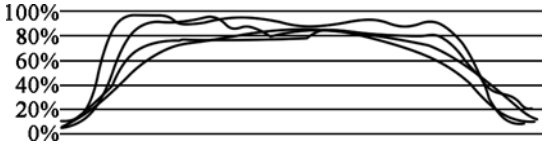


**Figure 3.** Utilization of CPU resources with VM migration

## 4. Conclusion

This paper presents the taxonomy of server resources for the dynamic server resource management.

We proposed a redistribution method between server resources using VM migration based on OpenStack.

It can significantly reduce the power consumption and provide server load balancing of data centers.

In the future in order to enhance the speed of VM migration, the threshold must be decided precisely and not only CPU utilization but also memory and network bandwidth utilization must be considered to calculate load values.

## Acknowledgements

We gratefully acknowledge the help provided by and constructive comments of the anonymous referees.

## References

1. Ardagna D, Panicucci B, Trubian M, Zhang L, 2012. Energy–aware autonomic resource allocation in multitier virtualized environments, *IEEE Transactions on Services Computing (TSC)*, 5(1), 2–19.

2. Antonio Corradi, et al, 2014. VM Consolidation: A real case based on OpenStack Cloud, *Future Generation Computer Systems*, 118–127.

3. Beloglazov A, Abawajy J, Buyya R, 2011. Energy–aware resource allocation heuristics for efficient management of data centers for cloud computing, *Future Generation Computer Systems* (FGCS), 28(5): 755–768.

4. Feller E, Rilling L, Morin C. Snooze, 2012, A scalable and autonomic virtual machine management framework for private clouds, *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 482–489.

5. Juliano Araujo Wickboldt, Rafael Pereira Esteves, Márcio Barbosa de Carvalh, Lisandro Zambenedetti Granville, 2014, Resource management in IaaS cloud platforms made flexible through programmability, *Computer Networks*, 68, 54–70.

6. Md. Mahbub–E–Noor, et al. 2014. Evaluation of OpenStack (Havana Release) and CloudStack (4.3 Release) Open Source Cloud Solutions, *Information Technology Journal*, 2 508–2 513.

7. Speitkamp B, Bichler M, 2010. A mathematical programming approach for server consolidation problems in virtualized data centers, *IEEE Transactions on Services Computing (TSC)*, 3(4), 266–278.

8. Thomas Szyrkowiec, et al. 2015, First field demonstration of cloud datacenter workflow automation employing dynamic optical transport network resources under OpenStack and OpenFlow orchestration, *OPTICS EXPRESS*, 41–48.

9. Wang X, Wang Y, 2011, Coordinating power control and performance management for virtualized server clusters, *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 22(2), 245–259.

10. Yoji Yamato, 2015, OpenStack hypervisor, container and Baremetal server's performance comparison, *IEICE* 21. *Communications Express*, 228–232.