

An Automatic Network Live Migration Mechanism for Cloud Computing

Min-Xiou Chen

Department of Computer Science and Information
Engineering,
National Dong Hwa University,
Hualien, Taiwan, R.O.C.
E-mail:mxchen@mail.ndhu.edu.tw

Huai-Hsien Terng

Department of Computer Science and Information
Engineering,
National Dong Hwa University,
Hualien, Taiwan, R.O.C.
E-mail: loui921.terng@gmail.com

Abstract—The Cloud Computing is the most popular topic at present that allows creating their services on a virtual network environment. The migration mechanism was proposed to provide services, computing jobs and resources moved from a virtual environment to another virtual environment, and service providers can fit out their logic network architectures to offer services for user requirements. In the paper, the migration mechanisms were classified into host migration, resource migration, and network migration. The network migration mechanism can move the set of services and resources from a set of virtual/real machines to another set of virtual/real machines and reconfigure the network configuration automatically based on the migrated network environment. To achieve this, the proposed mechanism is implemented based on Open vSwitch and XEN, and integrates the existing network protocols. In our experimental results, we confirm that the proposed mechanism can provide live network migration.

Keywords- Virtualization, Virtual Network, Migration, Network Migration, Cloud Computing, Software Defined Network.

I. INTRODUCTION

Over the past few years, the Cloud Computing is the most popular topic that allows creating users services on a virtual network environment. Such services can be grouped in to IaaS (Infrastructure as a Service), PaaS (Platform as a service), and SaaS (Software as a service). IaaS is the most applicable service at present, such as Amazon EC2 [1] is a well-known IaaS provider. Base on IaaS, people can rent an instance from the IaaS provider and setup a system and install software.

According to the definition [2], Cloud Computing provides a ubiquitous, convenient, and on-demand network access with shared computing resources which can be rapidly provisioned and released according to the user requirements. The IaaS is developed based on virtualization technology to offer virtual machines for user requirements. Virtualization technology can provide a logical representation of a hardware platform, operating system, storage device, or network resources.

The network virtualization is one important topic of virtualization technologies. It involves hardware virtualization and combines with resource virtualization, including bandwidth, buffer space, queues and network functionality into a single, software-based administrative entity. It also provides a virtualized network addressing space within or across the network subnets. Based on the

definition proposed in [2][3][4], the network virtualization technology should quickly allocate network resources from the common resource pool, create network path and configure network environment according the requirements. Within the network virtualization technology, IaaS providers can fit out their logic network architectures to offer services for the user requirements.

Moreover, these services, computing jobs or resources may need to move from a virtual environment to another virtual environment according the user requirements. The migration mechanism was proposed to provide services and resources migration between two virtual environments. Most of the migration mechanisms [5][6][7] focus of how to provide a live migration mechanism to move the services and resources from a virtual/real machine to another virtual/real machine. The VMware [8] proposed a solution to provide a set of resources move from a virtual environment to another virtual environment.

However, these services and resources may deployed on several virtual machines and are often tightly coupled with the underlying network environment, that needed to add some configurations to support the QoS of these services and resources. In the paper, we focus on the network migration mechanism which can move the set of services and resources from a set of virtual/real machines to another set of virtual/real machines and reconfigure the network configuration automatically based on the migrated network environment. Some shell scripts and existing network protocols were integrated with Open vSwitch [9] and XEN [10] in order to provide network migration without installing any middle ware.

The remainder of this paper is organized as follows. The problem statements, related literatures, projects and protocols are discussed in Section 2. The proposed approach is described in Section 3. Section 4 presents the performance evaluation for the proposed system, and conclusions and suggestions for future directions are given in Section 5.

II. RELATED WORKS

Cloud computing deeply depends on network communication, and must be provided high quality and stable communication for a consumer to access services through the Internet. According the final definition proposed in [2], cloud computing is a technology to provide a ubiquitous, convenient, and on-demand network access to a shared computing resources which can be rapidly deployed and reconfigured according to the user requirements and system manager interaction. In order to provide resource

rapidly deployed and reconfigure according to the demand, virtual machine (VM) migration was proposed to move services and resources from one location to another. A lot of researches and projects [5]-[8][11][12] were proposed to provide VM migration. Most of these proposed mechanisms needed to install their software and hardware in order to support migration.

Some of the mechanisms were designed for **off-line migration** which needed to block the services during the migration, and the others can provide **live migration** that the service can still be accessed during the migration. The **off-line migration** is a stop and copy mechanism, that will stop the current services, copy the configuration or image file to the target machine and restart the service. The **live migration** can still provide services during the migration. It's obviously that the architecture of live migration mechanism is more complete than that of off-line migration mechanism. According to the VM migration serviced scale, these VM migration mechanisms can be classified into three types:

- Host migration:** this mechanism can move a set of VMs and resources hosted on a real machine to another real machine. Most the VM projects [5]-[8] or VMs can support host migration.
- Resource migration:** this mechanism can move a set of resources which are deployed on a set of VMs to another set of VMs. VMware can provide a set of resources move from a virtual environment to another virtual environment, but these machines needed to install VMware.
- Network migration** [11][12]: this mechanism can move a set of VMs and resources which are deployed on a set of real machines to another set and the target network can be reconfigured according to the original network configuration.

Moreover, based on the mapping model between the migrated machine and targeted machine, these migration mechanisms can be grouped into **symmetrical migration** and **asymmetrical migration**. The **symmetrical migration** means that the VMs or resources hosted on one machine should be moved to one machine, and the **asymmetrical migration** means that the VMs or resources hosted on one machine can be moved to more than one machine. It's obviously that the host migration is symmetrical migration, and resource migration and network migration are asymmetrical migration.

III. SYSTEM ARCHITECTURE

The proposed architecture is shown in the Figure 1. The gateway is a router, and a switch will connect several real machines. The Open vSwitch and XEN are installed in these real machines. The Open vSwitch was introduced to provide switch services for the VMs between a set of real machines. The DNS server and DHCP server are hosted on the software router. The DNS server will keep the relation between the IP and hostname, and the DHCP server can be used to provide the automatic host deployment mechanism proposed in [13]. FTP was introduced to transfer the VMs' files and a FTP server is used to store the VMs' files.

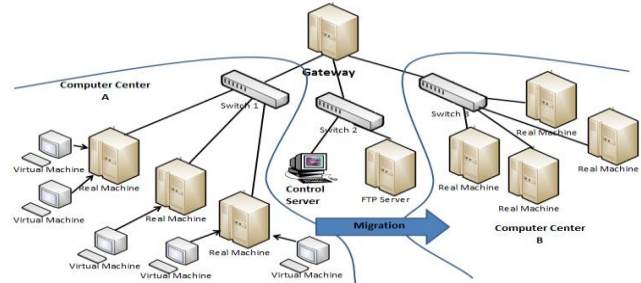


Figure 1. The Proposed Architecture.

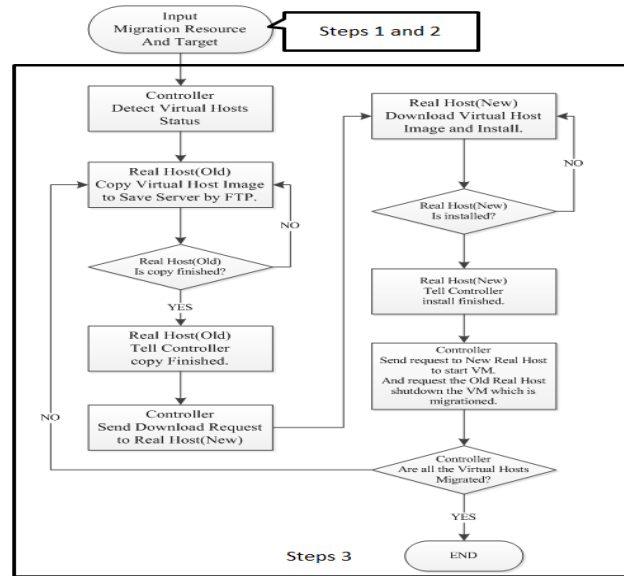


Figure 2. The Migration Flow Chart.

Based on the Software Defined Network (SDN) proposed in [14], a centralized controller was designed to manage the network state, to exchange control messages between network components (such as routers, switches and network servers) and to provide some methods to control these network components. According to the definition of SDN, in our proposed architecture, the control server used to collect the information of these migrated VMs and resources, and to select the target machine. Some scripts are implemented in order to control the migration scheduler. The control server also performs the network setup procedure to update the network configuration.

The migration flow can be divided into three steps as the following and shown in the Figure 2:

- Resource information collection:** In the step, the control server will call the scripts to collect the information of migrated VMs and resources, such as CPU utilization, memory/storage utilization, bandwidth requirement, and so on.
- Target machine decision:** In the step, the control service will select target machine for each migrated VM and resource, and arrange the migration order of each migrated VM and resource.
- Virtual machine/resource migration:** In the step, the control server will perform the scripts to back up the migrated VMs and resources, transfer the backup file to the FTP server, and transfer the backup file to the

target machine. The network configure procedure will be performed after migration.

The detail migration flow of proposed mechanism will be shown in the following sections.

A. Resource Information Collection

In the first, the users need to submit a migrated machine list to the centralized controller. The centralized controller may remote login the migrated machine through ssh and call the shell script to collect the required resource information according to the list. For example, df can be used to collect the disk utilization, “ps -r” and “ps -m” can be used to collect the CPU utilization and memory utilization, respectively. The centralized controller also gets these machines’ network configuration. The information will be stored on a file, and the target machine decision will select the target machine based on the information.

B. Target Machine Decision

The major purpose of target machine decision is to select the target machine for each migrated machine. In the symmetrical migration, the amount and capacity of target machines should be the same as that of the migrated machines. Thus, the target machine decision is very simple, and the centralized controller moves these VMs from the migrated machine to the target machine one by one. The result of the target machine decision is the migration order. There are two kinds of migration order in the symmetrical migration. The first order is to select one migrated machine, move these VMs to a target machine, and select next one, and so on. The second order is similar to round-robin. It selects a VM from one migrated machine, moves the VM to a target machine, and selects a VM from another migrated machine, and so on. The service break time of second one may be better than that of first one.

In the asymmetrical migration, the amount and capacity of target machines should be larger or equal to that of the migrated machines. The target machine decision is more complicated than that of the symmetrical migration. In the proposed mechanism, the migration order is decided based on the following algorithm. Let N be the set of the migrated VM, and M be the set of the target machines. Let L be the migration order list, and N_L be the set of the VM in L . $REQ(vm1)$ denotes the resource requirement of VM $vm1$, and $CAP(m1)$ denotes the capacity of real machine $m1$. More than one VM can be hosted on a real machine, and N_{m1} denotes the set of VM hosted on real machine $m1$. Let assume that the $REQ(N)$ should be less than or equal to $CAP(M)$.

1. Selects vm , and $vm \in \{N\} - \{N_L\}$
2. Selects $m \in \{M\}$, and $CAP(m) \geq REQ(vm) + \sum_{i \in N_m} REQ(i)$.
3. Inserts vm into L .
4. If $\{N\} \neq \{N_L\}$, return step 1. Else, the L is the final migration order.

The proposed mechanism can support both the symmetrical and asymmetrical migrations.

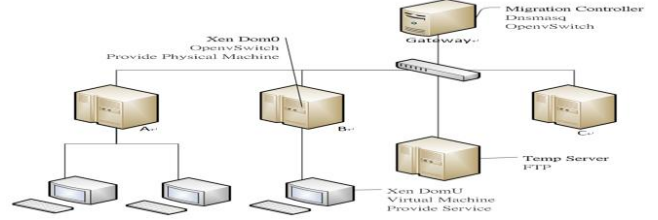


Figure 3. The Experiment Environment.

C. Virtual Machine/Resource Migration

The migration order will be generated in the previous step, the control server will perform the script codes according the migration order to back up the migrated VM or resources into an image file, and transfer the image file to the FTP server. When the image file is stored at the FTP server, the control server will command the target machine to transfer the image file, and restart the VM and resources at the target machine. In order to provide live migration, in the paper, the migrated VM will use the same IP in the target machine, but the MAC address of the migrated VM may be changed. The ARP cache of these machines and router will be updated after the migration.

IV. PERFORMANCE EVALUATIONS

The major objective of the proposed mechanism is to provide the network migration using existing network protocols. As shown in the Figure 3, there are five real machines in the experiment to show the performance of the proposed mechanism. The control server is hosted on the migration controller, and the DNS server, DHCP server, and OPEN vSwitth are installed at the control server. The FTP server is hosted on the temp server. The machines A, B and C were used to provide VM or migration.

There are five scenarios in the experiments. In the scenario 1, there are two VMs hosted on the machine A, and one VM hosted on the machine B. The VMs on the machine A will be migrated to the machine C, and then the VM on machine B will be migrated to the machine C. In the scenario 2, the VM on machine B will be migrated to the machine C first, and then the VMs on machine A will be migrated to the machine C. In the scenario 3, there are two VMs hosted on the machine A, and one VM hosted on the machine C. The VMs on the machine A will be migrated to the machine B, and then the VM on the machine C will be migrated to the machine B. In the scenario 4, the VM on the machine C will be migrated to the machine B first, and then the VMs on the machine A will be migrated to the machine B. In the previous scenarios, all the configuration of VMs are the same, and each VM may provide HTTP services. The scenario 5 is the same as the scenario 1, but the VCPU of VM1 is 2.

The major objective of the experiments is to show the live migration of the proposed mechanism. The response time can be used to show the performance of the proposed mechanism. A machine deployed at the outer network accesses the HTTP service, and the response time are shown in the Figures 4 to 8. The response time of VM1 is denoted as solid line, the VM2 is denoted as dashed line, and the VM3 is dotted line.

The response time of these scenarios are shown in the Figures 4 to 8, respectively. There is one peak at each line. The peak is the break time of migration. The Figures 4 and 5 show that the break time of these VMs during migration are between 18 to 19 seconds. The Figures 6 and 7 show that the break time of these VMs during migration are about 25 seconds. The hardware specification of C is better than that of B. These results show that the hardware capacity will affect the service break time when the VMs or service migrated. The Figure 8 shows the response time of scenario 5. It is obviously that the service break time of VM1 in scenario 5 is slightly larger than that in scenario 1. The results show that the image complexity of VM will slight affect the service break time.

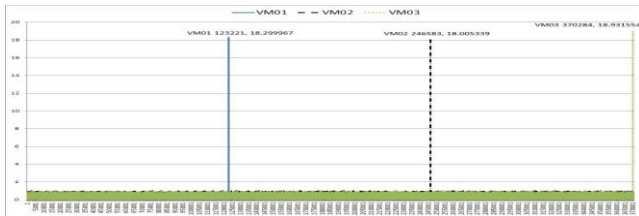


Figure 4. The Response Time of Scenario 1.

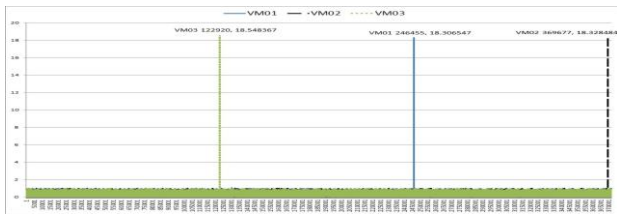


Figure 5. The Response Time of Scenario 2.



Figure 6. The Response Time of Scenario 3.

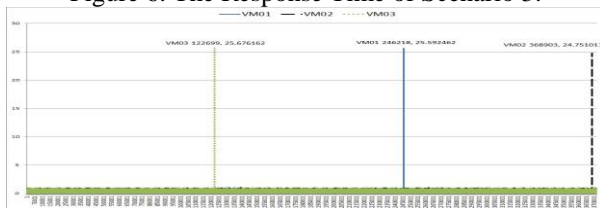


Figure 7. The Response Time of Scenario 4.

V. CONCLUSIONS

The migration mechanism was proposed to provide services, computing jobs and resources moved from a virtual environment to another virtual environment. In the paper, some shell scripts and existing network protocols were integrated with Open vSwitch and XEN in order to provide the network migration mechanism which can move a set of

services and resources from a set of virtual/real machines to another set of virtual/real machines and reconfigure the network configuration automatically based on the migrated network environment. The performance evaluations show that the proposed mechanism can move the set of services and resources between different set of virtual/real machines, and the service break time could be less than on minute. In the future, the target machine decision will be improved in order to provide better migration, and the virtual machine/resource migration will be enchanted to move the VM and resources across the router.

ACKNOWLEDGMENT

This work was supported by the National Science Council of Taiwan, R.O.C. under Grant NSC101-2628-E-259-004-MY3.

REFERENCES

- [1] Amazon EC2. <http://aws.amazon.com/ec2/>.
- [2] P. Mell, and T. Grance, "A NIST National Definition of Cloud Computing, version 1.5", National Institute of Standards and Technology (NIST), Oct 2009.
- [3] N. M. Mosharaf Kabir Chowdhury, Raouf Boutaba, "Network virtualization: state of the art and research challenges", IEEE Communications Magazine, Vol. 47, No. 7, pp. 20-26, July 2009.
- [4] T. Anderson, L. Peterson, S. Shenker, J. Turner, "Overcoming the internet impasse through virtualization", Computer, Vol.38, No. 4, pp. 34-41 April 2005.
- [5] "Virtual Machine Live Migration Overview", <http://technet.microsoft.com/zh-tw/library/hh831435.aspx>
- [6] "KVM Migration", <http://www.linux-kvm.org/page/Migration>
- [7] "XEN Migration", <http://wiki.xen.org/wiki/Migration>
- [8] "VMware", <http://www.vmware.com/>
- [9] "The Open vSwitch, a production quality, multi-layer open virtual switch", Open vSwitch, <http://openvswitch.org/>
- [10] "Xen hypervisor", the powerful open source industry standard for virtualization, Citrix Systems, Inc., <http://xen.org/products/xenhyp.html>
- [11] Eric Keller, Soudeh Ghorbani, Matt Caesar, Jennifer Rexford, "Live migration of an entire network (and its hosts)", HotNets-XI Proceedings of the 11th ACM Workshop on Hot Topics in Networks, pp. 109-114, 2012.
- [12] Yi Wang, Eric Keller, Brian Biskeborn, Jacobus van der Merwe, Jennifer Rexford, "Virtual routers on the move: live router migration as a network-management primitive", SIGCOMM '08 Proceedings of the ACM SIGCOMM 2008 conference on Data communication pp. 231-242.
- [13] Min-Xiou Chen, and Kuo-Le Mei, "Mechanism of Automatic Deployment for Virtual Network Environment", 2013 42nd International Conference on Parallel Processing.
- [14] "SDN/OpenFlow", <http://www.nec-labs.com/~lume/sdn-reading-list.html>

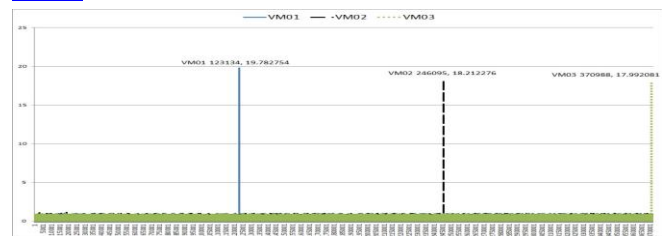


Figure 8. The Response Time of Scenario 5