Live Migration Based on Cloud Computing to Increase Load Balancing

Jananta Permata Putra[1], Supeno Mardi Susiki Nugroho[1,2], Istas Pratomo[1]
Department of Electrical Engineering[1], Department of Computer Engineering[3]
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
jananta15@mhs.ee.its.ac.id, mardi@its.ac.id, istaspra@ee.its.ac.id

***Abstract*—Nowadays, the availability of Internet services is very important. One of these internet services is information system service. When the information system can not be accessed, the user cannot access the information. Therefore, the server used for that application should remain online nonstop. However, in reality the services failure often happens due to server error or repairing process. One of the solution is cloud computing so that the running application can be moved to another server known as migration. Due to regular migration process takes time to move the data, so the services cannot be access. It is necessary to migrate directly to the destination server without moving the data. One way to implement that live migration is replicating the existing data on the origin server to the destination server at any time. Direct migration can be done by replicating data from the origin server to the destination server every time. In this paper we analyze performance of the live migration on cloud computing to improve the availability of information systems. The goal is for users to access information services continuously.**

Keywords—Cloud Computing, Virtual Machine, Live Migration

I. INTRODUCTION

Now a day the access of Internet services is very important. One of frequently used Internet service is the information system service. If that services are not accessible, then the people who use those services will not be able to find the latest information provided by that information system. The information system services definitely need server to run an application which serve that information systems. If that servers are down, then automatically the services that provided will also be stopped. So slightest time the services provided stopped will make an impact on those who need it.

One of the ways to prevent the service provider stopped is using virtual machine. Virtual Machine is a method to create multiple virtual servers in one physical server. One of the advantages of the virtual machine is able to move a virtualized server to another physical server if the used server have problems or still in maintenance. These capabilities are usually called migration. At regular migration process (Cold Migration), the virtual machine must be turned off to prevent corrupted data when on hard drive when move. If the virtual machine moving process are in position of virtual servers is on then the process is called live migration. The time required to perform regular migration is proportional to the storage capacity of the virtual machine that used. So, it still takes time to turn on a virtual machine that is used. To minimize the time required for migration is using a way that be able to copy the process and data from one server to another server. The way is to use storage media can be accessed in both the server or in other word is shared storage.

To increase High Availability (HA) on a server, the server must remain power on in any circumstances. If a physical server is experiencing problems or maintenance process, then the server service had to be moved to another place. In the Virtual Machine, if the data from the guest VM is stored in the local storage and want to migrate the vm to another host it will requires power off the guest and then it will can be moved. After being moved, it will require a certain time for the guest to power on again. Because of that the High Availability of such services will be reduced. Therefore, it is necessary to do live migration of virtual machines. One of the requirements for live migration is to allow the origin server and the destination server to access the same storage media. To prevent physical servers do not experience excessive load it will require live migration that can adapt to the server performance of the server.

In the previous paper there was some research on cloud computing. For example, Cloud Performance Modeling and Benchmark Evaluation of Elastic Scaling Strategies from Kai Hwang, Xiaoying Bai, Yue Shi, Muyang Li, Wen-Guang Chen, Yongwei Wu who evaluated the benchmark of Cloud [7]. In previous research about live migration in the cloud computing with the title Pacer: A Progress Management System for Live Virtual Machine Migration in Cloud Computing, the authors solve management problems with accurately calculate the predicted time of migration and use those predictions to get better management of migration [2]. Another research is "iAware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud" that designs a simple multi-resource demand query mode to estimate and minimize migration and interference between VM guest [11]. Research conducted by J. Octavio

Gutierrez-Garcia is using agency collaboration to balance varying load. When the host has the burden of load, the agents will cooperate with each other to extract data host and provide the best migration destination in accordance with the given rules [3]. In research by Yonghui Ruan, Zhongsheng Cao, and Zongmin Cui, entitled Efficient and Self-Adaptive Live Migration of Virtual Machines, they make improvisation of the pre-filter-copy (PFC) algorithm to handling iterations in memory migration that uses an algorithm pre-copy. The PFC algorithm has objective to reduce migration time and bandwidth consumption of pre-copy algorithm with same downtime time [1]. In this research will make load management system to balance servers load so it will increase high availability of information system service without having to make the server has excess load.

II. DESIGN SYSTEM

A. Miniature Cloud Design

The design of the network topology that will be used as follows

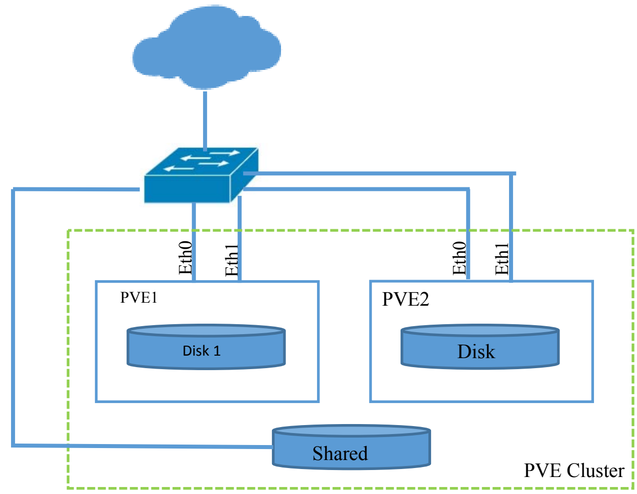

Fig. 1. Miniature Cloud Design

Designs to be created will be using two physical servers. Inside each server there are two disks are mounted. The first disk is a local disk that contained in the physical server. That disk is used for PVE installation and the second disk is a Network Attached Storage (NAS) that shared to both servers. In the design is also have two computers networks. The first network is made to access a virtual machine while the second network created for the lines of communication between the server and the NAS. That network is separated so the communication network in the virtual machine is not disturbed by the flow of data on the NAS.

The OS will be installed on the server Proxmox Virtual Environment (PVE). The installation is boot to the ISO file that has been provided, then follow the wizard such as hostname and IP address for PVE management. We use book from Rik Goldman, "Learning Proxmox VE" to understand how PVE works.To be able to migrate between the two servers, the two servers must be in the same management or in other words in a single cluster. We use book from Simon M.C. Cheng, "Proxmox High Availability" to configuration cluster. NAS is used as shared storage for the guest storage media. Shared storage on PVE is needed in order to perform live migration.

B. Live Migration Scenario

Live migration works by copying the memory between the two hosts. Once all the memory is completed copied then the destination host can access the files in storage. When the load performance is not the same on both hosts, the guest will move from one server which did extensive work to the host that has a lighter work. The migration performed live so that the guest will not off. The Performance which measured is CPU, memory, and more.

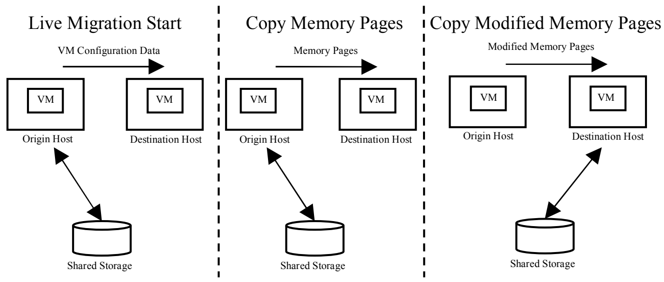

Fig. 2. Live Migration Step

Live migration steps are as follows:

1. The VM Configuration are copied from originating host to destination host.
2. Originating host Memory is copy to destination host.
3. The originating host memory page that change will be marked and copied to the destination host.
4. The process of copying the memory pages will immediately continue until the memory page turns into a small.
5. VM guest process will be paused at the origin server.
6. The memory page is copied to the destination host for the last time.
7. VM Guest process will continue at the destination host.

8. ARP (Address Resolution Protocol) routing table will be updated on the network so that the host can communicate with the network.

Measurement Scenarios

The first measurement is to compare the duration of migration between using live migration and cold migration. The experiment will be repeated with difference capacity of the guest disk. Existing data on guest are made same then guest will be migrated with live migration or cold migration. Then the capacity of the disk will be added and migrated again just like the previous step.

The second measurements are performed on both hosts when before migration, during migration, and after migration. Measurement using *sar command* from systat package. Measurements are also in the migrated VM. Two VMs are created in one host so the load between two hosts will be different, since one host runs two VMs while the other hosts do not run any VM. Then we run *sar command* with to know the average of CPU process in 10 seconds. The command works for 90 seconds. When the command is executed, one VM will move without having to die to another host. The input of the measurement is to move the VM from the origin host to the destination host. By using *sar command*, we will get the CPU performance statistics.

III. Result

The first result we will compare between live migration and migration with different guest disk size.

TABLE I. Comparision Between Cold Migration and Live Migration

VM Guest Disk Size (GB)	Duration (seconds)	
	Cold migration	Live Migration
8	69	10
9	74	10
10	77	11
11	78	10
12	81	10
13	87	10
14	89	10
15	94	10
16	100	10

From the Table I above it can be seen that with cold migration then the time required for migration will increase if the disk size of the guest is added. While in live migration, the time required for migration is relatively similar.

Fig. 3 shows a comparison between cold migration and live migration from Table I.

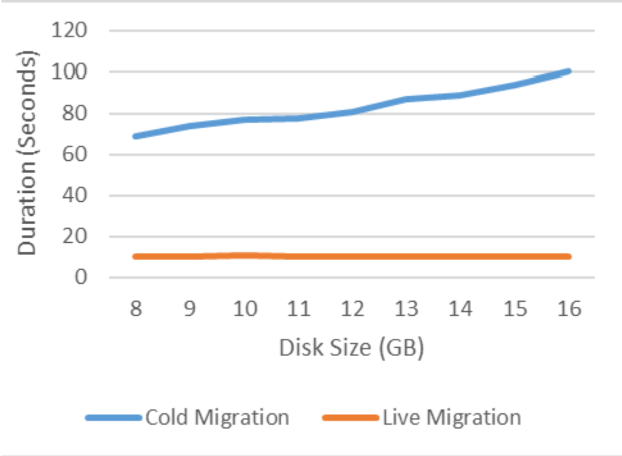

Fig. 3. Comparison between Cold Migration and Live Migration

The second results measurement of this study were conducted by looking at the CPU usage of both hosts by using sar command. Table I shows the comparison of the CPU usage percentage between two hosts with a 10 second pause for 90 seconds on both hosts.

TABLE II. Comparision of CPU Host Usage Percentage

Time	antapx1 CPU process (%)	antapx2 CPU process (%)
20:10:59	1.41	0.55
20:11:09	1.46	0.35
20:11:19	1.43	11.72
20:11:29	5.56	3.93
20:11:39	2.53	0.73
20:11:49	1.03	0.58
20:11:59	0.93	0.85
20:12:09	1.16	0.83
20:12:19	0.93	2.21

Table II above describes the comparison of CPU processes in both hosts, at 20:11:19 the process on antapx2 rises drastically. Before migration load on antapx1 host is greater than antapx2. Meanwhile, after migration the CPU load of antapx1 CPUs decreases and CPU load on antapx2 increases. Fig. 4 shows a comparison graph between antapx1 and antapx2 hosts based on Table I.

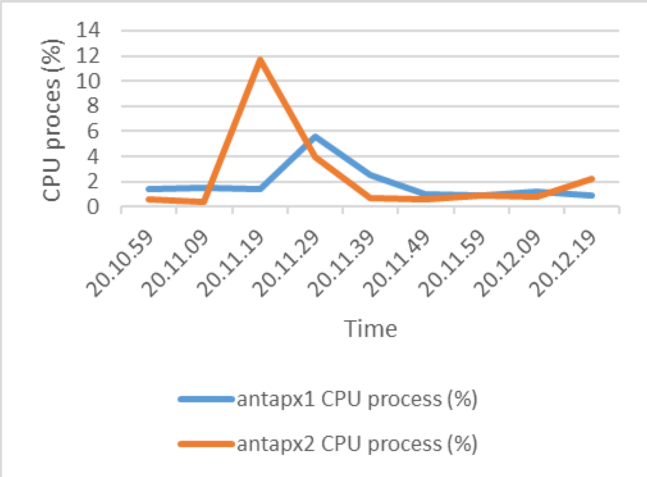

Fig. 4. CPU process comparison between two hosts

While the following is migration step when vm start migration and its time.

20:11:21 starting migration of VM 1020 to node 'antapx2' (10.199.13.38)

20:11:21 copying disk images

20:11:21 starting VM 1020 on remote node 'antapx2'

20:11:24 start remote tunnel

20:11:25 starting online/live migration on unix:/run/qemu-server/1020.migrate

20:11:25 migrate_set_speed: 8589934592

20:11:25 migrate_set_downtime: 0.1

20:11:25 set migration_caps

20:11:25 set cachesize: 214748364

20:11:25 start migrate command to unix:/run/qemu-server/1020.migrate

20:11:27 migration status: active (transferred 168054758, remaining 17330176),

20:11:27 migration xbzrle cachesize: 134217728 transferred 0 pages 0 cachemiss

20:11:29 migration speed: 512.00 MB/s - downtime 38 ms

20:11:29 migration status: completed

20:11:33 migration finished successfully (duration 00:00:13)

IV. Discussion and Conclusion

In this section will be discussed about the analysis and conclusions of the results obtained. In the first measurement, we can see that if using regular migration or cold migration it will take longer than live migration. It happens because migration also moves data from local disk host origin to local disk destination host. In cold migration the larger the disk size of the guest, the greater the duration required for migration. Whereas in the live migration the duration required for migration is relatively same. It happens because only move the memory pages. Memory pages in guest are almost same because the guest vm providing the same service.

The next measurement is to measure the workload on the host. In Table II the process of CPU usage between two hosts is found, initially load CPU on antapx1 has a larger load than antapx2 because it runs two guest VM at once, while CPU usage at antapx2 host is not higher than antapx1 because it still not running any guest VM. Then, at 20:11:19 antapx2 CPU usage rises from before, whereas at 20:11:29 antapx1 the CPU usage goes up. This happens because at that time the live migration is running so it requires CPU process to run it. Once the migration process is complete then CPU usage in both hosts will decrease. When viewed from time before migration, CPU usage of both hosts has different loads. Whereas after migration, CPU load at both hosts become almost the same.

From above analysis, it can be concluded:

1. In ordinary migration, the larger data on disk is migrated, the greater time it takes for migration.
2. Live migration takes less time than regular migration.
3. Live migration can easily balance server loads without disturbing running guest VM.

We hope that this paper can give the detail information how to balance the CPU performance load between source host and target host in a virtualized environment cluster so that the system can split load in the cluster. By dividing the load, the host server will be able to maximize server performance. With high performance of server, the service provided by the server will not have any constraints so it can contribute to make user comfort using that services.

References

[1] Yonghui Ruan, Zhongsheng Cao, Zongmin Cui, "Pre-Filter-Copy: Efficient and Self-Adaptive Live Migration of Virtual Machines", IEEE Systems Journal, 2016

[2] Jie Zheng; T. S. Eugene Ng; Kunwadee Sripanidkulchai; Zhaolei Liu, "Pacer: A Progress Management System for Live Virtual Machine Migration in Cloud Computing" IEEE Transactions on Network and Service Management, 2013

[3] J. Octavio Gutierrez-Garcia; Adrian Ramirez-Nafarrate, "Collaborative Agents for Distributed Load Management in Cloud Data Centers Using Live Migration of Virtual Machines" IEEE Transactions on Services Computing, 2015

[4] Beom Heyn Kim, "Automated Virtual Machine Replication and Transparent Machine Switch Support for An Individual Personal Computer User", University of Toronto, 2011

[5] Kai-Yuan Hou, Mustafa Uysal, Arif Merchant, Kang G. Shin, Sharad Singhal, "HydraVM: Low-Cost, Transparent High Availability for Virtual Machines", University of Michigan, 2011

[6] Patrick Raad, Stefano Secci, Dung Chi Phung, Antonio Cianfrani, Pascal Gallard, and Guy Pujolle "Achieving Sub-Second Downtimes in Large-Scale Virtual Machine Migrations with LISP", IEEE Transactions on Network and Service Management, 2014

[7] Kai Hwang, Xiaoying Bai, Yue Shi, Muyang Li, Wen-Guang Chen, Yongwei Wu, "Cloud Performance Modeling and Benchmark Evaluation of Elastic Scaling Strategies", IEEE Transactions on Parallel and Distributed Systems, 2015

[8] Rik Goldman, "Learning Proxmox VE", Packt Publishing Ltd, 2016

[9] Simon M.C. Cheng, "Proxmox High Availability", Packt Publishing Ltd, 2014

[10] "Enhanced Live Migration of Virtual Machine Using Comparison of Modified and Unmodified Pages", International Journal of Computer Science and Mobile Computing, 2014

[11] Fei Xu, Fangming Liu, Linghui Liu, Hai Jin, Bo Li, Fellow, Baochun Li, "iAware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud" , IEEE Transactions On Computers, 2014