

Performance Analysis of Live and Offline VM Migration Using KVM

Garima Rastogi

Department of Computer Science and Engineering, DIT University/Department, Dehradun, 248009, India
Email: garimaverma.research@gmail.com

Rama Sushil

Department of Information Technology, DIT University/Department, Dehradun, 248009, India
Email: ramasushil@yahoo.com

Abstract—Virtualization is a core technology used for the implementation of cloud computing. It increases the utilization of resources such as processor, storage, network etc. by collecting various underutilized resources available in the form of a shared pool of resources built through the creation of Virtual Machines (VMs).

The requirements in cloud environment are dynamic therefore there is always a need to move virtual machines within the same cloud or amongst different clouds. This is achieved through migration of VMs which results in several benefits such as saving energy of the host, managing fault tolerance if some host is not working properly and load balancing among all hosts. In this experimental study, effort has been made to analyze the performance of offline and live VM migration techniques with respect to total migration time and downtime of VM migration. Kernel-based Virtual Machine (KVM) hypervisor has been used for virtualization and a series of experiments have been carried out in computer service center of IIT Delhi on their private cloud Baadal. The experiment results show that downtime during live migration is very less in comparison to the offline migration while the total migration time is more in comparison to the offline migration.

Index Terms—Underutilized, Scalability, Portability, Fault tolerance, Total Migration Time, Downtime.

I. INTRODUCTION

Cloud Computing is the latest technology which helps organization in reducing infrastructural and computing cost by making sharing of resources possible. It helps organizations to achieve coherence and economies of scale such as utility computing. According to [1] “Cloud is a parallel and distributed computing system consisting of a collection of inter connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service level agreement (SLA) established through negotiation between the service provider and consumers.”

Cloud computing appears to similar to some of the analogous systems such as Autonomic Computing, Client

Server model, Grid Computing, Mainframe computers, Peer to peer computing, utility computing etc. but is essentially more powerful in terms of functionalities to share and utilize resources more efficiently and effectively.

There are five essential features of cloud computing given by NIST (National Institute of Standard Technology) – rapid elasticity, resource pooling, on demand self-service, broad network access and measured services. In this, resources are pooled at the centralized places called as data centers that are accessible from everywhere on demand.

For implementing cloud computing, we rely on virtualization which is a core technology for this. This technology diverted the industry perspective from the utilization of resources physical to logical. It creates an abstract layer over the actual hardware and software [2]. It emulates a physical machine in software to run multiple operating systems on single machine hardware. The main goal of virtualization is to utilize the maximum capacity of available resources such as processor, storage and network. By creating virtual machines, it collaborates multiple unutilized resources into a shared resource pool and utilizes them by performing different tasks simultaneously to fulfill multiple user demands. These resources can be scaled on virtual machines i.e. allocated dynamically [3].

There can be various types of virtualizations like -

- Application Virtualization – In this, application/s including operating system of host machine is moved to the virtual environment. It is a technology in which the application is present somewhere else but is accessed by the client computer. The application behaves same as the local application on the client system. For e.g. - VMWare Thinapp, Oracle secure Global desktop etc.
- Storage Virtualization – It provides a virtual storage environment by collecting or combining various physical storages. Through this, distributed storage is managed in such a way as if it is one consolidated storage. After this virtualization, the availability of storage increases because now the applications do not have limited or a specific

resource. The storage can be updated any time without affecting the performance of the application [4].

- Server Virtualization – In this, existing server is moved into a virtual environment i.e. hypervisor, which is hosted on a physical server [3]. The resources of server are hidden from clients and the physical server is divided into multiple virtual environments. Web server virtualization is one of the most popular examples of this technology used for providing low cost web hosting services.
- Hardware Virtualization – This virtualization makes hardware components of real machine as virtual components. This technology hides all the physical components and details of actual computing platform from end users.

Virtualization is done by using a hypervisor, a software, which acts as an intermediary between Virtual machine and physical hardware. It is used to create virtual machines. The hypervisor manages virtual hardware and guest Operating System on the said hardware on a virtual platform. Hypervisor can be native (Type-1) or Hosted (Type-2).

Type-1 hypervisor layer comes before the Operating System and runs on hardware directly to manage the guest Operating System [2]. This type of virtualization is known as full virtualization.

Type-2 hypervisor requires host Operating System to run it and the guest operating Systems are then managed by the hypervisor. This type of virtualization is called as Para-virtualization.

Fig. 1 is showing the placement of these hypervisors with hardware and operating system.

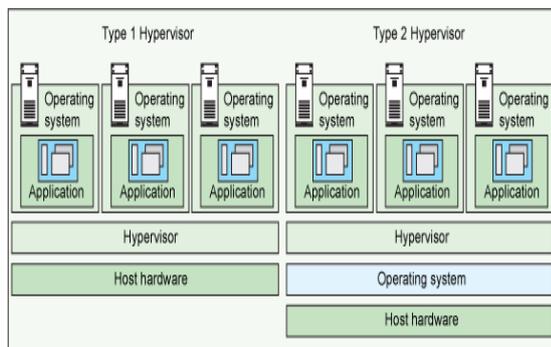


Fig.1. Type-1 and Type-2 Hypervisor

The increased utilization is achieved by collecting various underutilized or unutilized resources in the form of shared pool by creating virtual machines (VMs) to perform a number of different tasks simultaneously according to the demand of end users [5]. These resources can be allocated or de-allocated dynamically on VMs allowing a single physical host to be converted into number of virtual hosts. Each virtual host delivers a secure and isolated environment for applications. These environments can be customized in the form of software and hardware platform according to the demand.

There are certain situations when there is a need of migration of VMs from one host to another host. Some situations are as follows [6] –

A. Fault Tolerance

If one physical host faces some problems and does not run, tasks which are running on its Virtual Machines should not suffer. All VMs on the physical host will be migrated to some other host which is running properly.

B. Load Balancing

There can be a situation when one host has multiple VMs handling multiple tasks and there is another host which has no VM or less number of VMs with less number of tasks, then some VMs of overloaded host can be moved to under load host to balance the load.

C. Reduce Energy Consumption

In a situation where a host is running with less number of tasks or no tasks and some other host is running with more tasks, then all the tasks of first machine can be moved to the other machine and the former can be switched off to save power.

VM migration can be conducted offline as well as online (Live). In offline migration, a virtual machine is first suspended and then all configuration files are moved from “source host” to the “destination host”. After this, copied VM memory image is resumed at the destination. On the other hand, in live VM migration, we migrate running VMs from “source host” to “destination host” seamlessly. VM migration can have varying performance implications in offline and live mode.

In this research study, a series of experiments have been carried out to do performance analysis of VM migration using open source hypervisor KVM. The purpose to perform the analysis was to know if the reduction in the migration time was possible while VM was migrating. We migrated VMs in two modes live and offline- the results of the experiment showed that downtime during live migration was very less in comparison to the offline migration while the total migration time was more in comparison to the offline migration.

This paper is structured as follows: Section 2 describes the related studies, section 3 discusses the setup used in the experiment, section 4 includes the experiment outputs and the analysis of the Live VM migration and offline VM migration and the last section 5 describes the conclusion.

II. RELATED STUDIES

Some of important research studies that are related to carrying out performance analysis are mentioned as below:

In [4], the authors proposed a new Virtualization Techniques Model which can reduce the workload of a data center, complexity and costs also improved system capacity, flexibility, and efficiency. The model also claimed that it improves service Continuity.

In [5], the authors said that usage of large scale computing resources on cloud resulted in increased energy consumption. In this work they showed the migration of VM for the purpose of load balancing and proposed a method to minimize VM migration using load balancing for efficient utilization of resources. They designed a method to evaluate which VM should be migrated and where, in respect to improve the overall load balancing.

In [7], the authors performed downtime analysis of virtual machine migration through an experimental study to analyze the live migration downtime and the duration on the basis of memory load. They said it was very important to analyze the performance of VM during migration, especially its downtime to provide better Quality of Service (QoS).

In [8], authors developed an automated testing framework which measured the performance characteristics of live VM migration, such as total migration time, downtime etc. They applied this testing framework for analyzing the performance of memory migration and storage migration for various hypervisors like, Xen, and VMWare etc.

In [9], the authors said that live migration was a core feature of virtualization technology. They characterized few key parameters that affected the live migration, especially in Xen virtualization environment. They discussed the relationship between important parameters that affect live migration to conclude how performance migration could considerably depend on the workload. They also provided two simulation models to predict migration times with 90% accuracy.

In [10], the authors did a survey of various issues and techniques of VM live migration. They discussed the effects on performance metrics like total migration time, downtime etc. when the VM was migrated live on Wireless Network with heavy workload.

In a research by [11], the authors mentioned various benefits that could be achieved by the VM migration, such as high availability, fault tolerance, energy saving etc. They analyzed some parameters used during migration, such as total migration time, downtime; amount of data transferred empirically and proposed to choose right VM for migration.

In [12], the authors provided a model which could predict the performance of VM migration. They performed a series of experiments in different scenarios by taking a single VM, multiple VM with different memories and VM in parallel. The results of the experiment showed that parallel VM migration results in less aggregate migration time.

In [13], authors briefed the importance of virtualization technique and described the live VM migration technique. They evaluated the performance of TCP in live migration and concluded that the total migration time was increased if the VM size was increased.

In a research by [14], the authors explained about the pre-copy and post-copy VM migration techniques for memory intensive application. They discussed several techniques to provide better support for migration.

III. EXPERIMENT SETUP

This experiment was conducted in the Computer Service Center lab of IIT Delhi. The host machines of the Data Center were used which were originally created for their private cloud Baadal. The configuration of the host used was 2x4 core Intel ® Xeon ® CPU E5540 @ 2.53GHz and 12 GB RAM. Two hosts were used at the start. All the host servers could access the shared storage 50TB based on NetApp 3210V NAS and HP EVA6400 SAN with FC disks. For Virtualization KVM (Kernel Virtual Machine) as open source was used [15].

The experiment setup is shown in the Fig. 2 Two hosts of same configuration named Host1 and Host-2 were taken. Host-1 had a number of VMs on it with Operating System Ubuntu 12.04. In the experiment a number of migrations were carried out from host-1 to host-2 to analyze the performance. The concept of shared memory was used for storing the image of VMs and for VM migration between hosts; Secured Shell (SSH) was used. It is a network protocol which allows data exchange between two hosts on the network through a very secure channel. Although there are many protocols for data exchange like FTP, POP and Telnet but these are not secure as one can transfer information in the form of plain text, making it easy for hackers to access it. Therefore, SSH channel was used that could restrict hackers and attackers from hacking information. Hence it provided a safe and secure way to transfer data files.

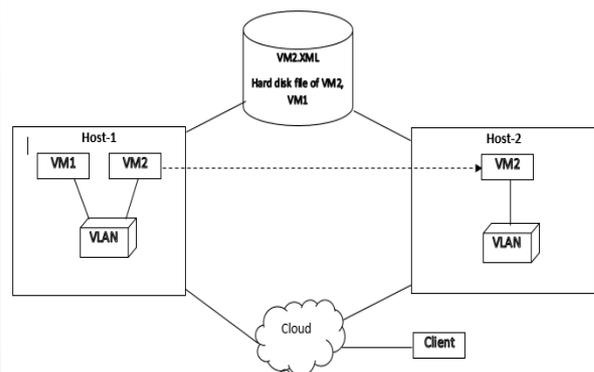


Fig.2. Experiment Setup

A. Steps Followed To Create VM

- VM on host1 was created using virt-install command. Fig. 3 shows the status of VMs on host-1 whether they were running or shutdown.

```
# ssh root@10.16.171.23 virt-install --virt-type kvm --
name          vm1          --ram          1024          --
cdrom=/home/downloads/ubuntu.iso          --disk
/var/lib/libvirt/images/vm1.qcow2,          size=14,
format=qcow2,bus=virtio -- network bridge=br0,
model=virtio, mac=52:54:00:10:55:ef --graphics vnc,
port=5901, listen=0.0.0.0, password=abc --noautoconsole
--os-type=linux
```

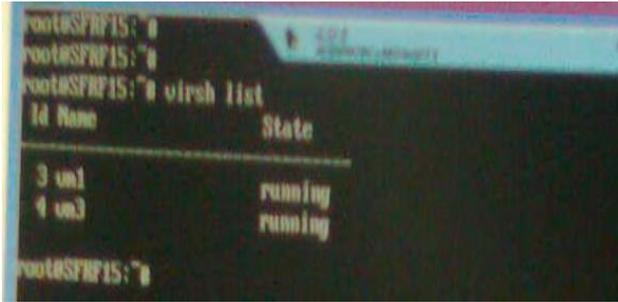


Fig.3. Status of VMs on host-1

As exhibited in the above figure, the state of VM3 and VM4 is shown as running after executing the above command.

For making changes in the interface file of the host-1, bridge port was added into it.

- Live Migration was performed of VM from host-1 to host-2

```
# ssh root@10.16.171.23 virsh --connect
qemu+ssh://root@10.16.171.9 migrate --live 10.16.171.23
qemu+ssh://10.16.171.23/system
tcp://10.16.171.23:48160
```

- Offline migration was performed of VM from host-1 to host-2

```
# ssh root@10.16.171.23 virsh -c qemu+ssh://
10.16.171.9 define /etc/libvirt/qemu/vm1.xml
# ssh root@10.16.171.23 virsh undefine vm1
```

Total 6 VMs were created for the experiment and were installed with Ubuntu 12.04 as operating system. The VMs shared the same disk read-only image which was stored on a host server called as the storage server and could be accessed through NFS4 (Network File System version 4). Changes to the virtual disks during VMs execution were stored in the redo log files.

Offline as well as online migration of all VMs was performed with a set of applications running in each of the experiments. The application included python, MySQL, Office etc.

B. Metrics for VM Migration

Two metrics were used for migration –

- Total Migration Time – It is total time when all states, CPU and memory were transferred [15].

$$\text{TMT (Precopy)} = \text{TI} + \text{OV} \quad (1)$$

$$\text{TMT (Postcopy)} = \text{OV} \quad (2)$$

- Downtime- It is time between VM Resume and Stop state.

$$\text{TDT (Precopy)} = \text{DM} + \text{VS} \quad (3)$$

$$\text{TDT (Postcopy)} = \text{VS} \quad (4)$$

Where

TI- total time of all iterations, OV- Overhead, DM - Dirty Memory, VS - VM State Time

IV. ANALYSIS AND OUTPUT

A. Offline Migration Analysis & Output

Offline migration had three steps viz. Suspend, Copy and Resume. In the 1st step, the VM was suspended at the source host (host-1) and its memory was captured to VM memory state file.

```
# ssh root@10.16.171.23 virsh suspend vm1
```

In the 2nd step, VM memory state file, configuration file and redo files were moved to the destination host (host-2) from the source host (host-1) in the 3rd step, the VM at destination restored its memory state and resumed the execution.

In the experiment all three steps during VM migration were analyzed. VMs were migrated sequentially one by one from host-1 to host-2. All the VMs had different configuration and different applications were running on them. It was analyzed what kind of impact would be there on migration time if different memory size were used for VMs as VM memory state file had a major role in migration process.

As exhibited in the plots drawn between time and memory size for the migration of 6 VMs offline as shown in Fig. 4, 5 and 6, the maximum time was taken by the VMs in copy phase(7.3seconds) in comparison to suspend (2.8seconds) and resume phase(0.7 seconds).

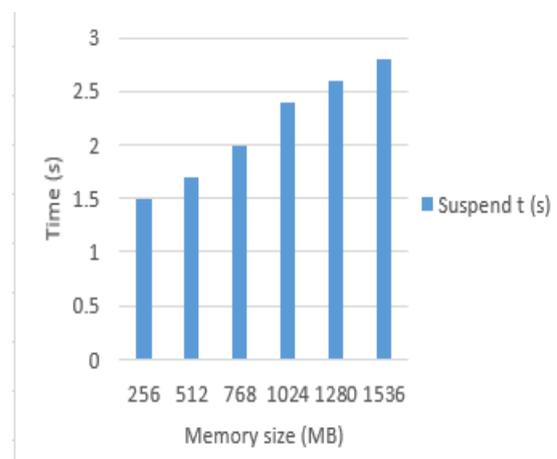


Fig.4. Plot of Suspended phase

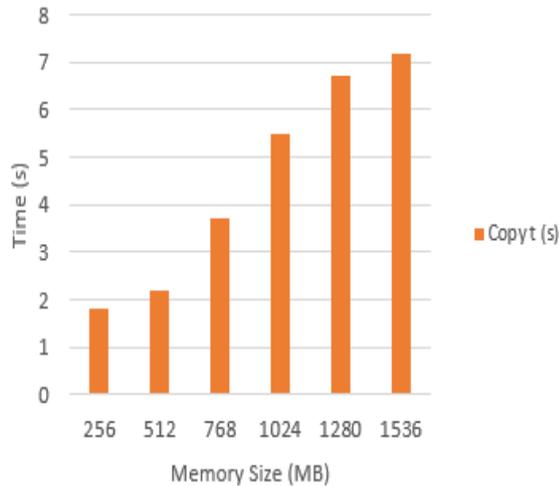


Fig.5. Plot of Copy phase

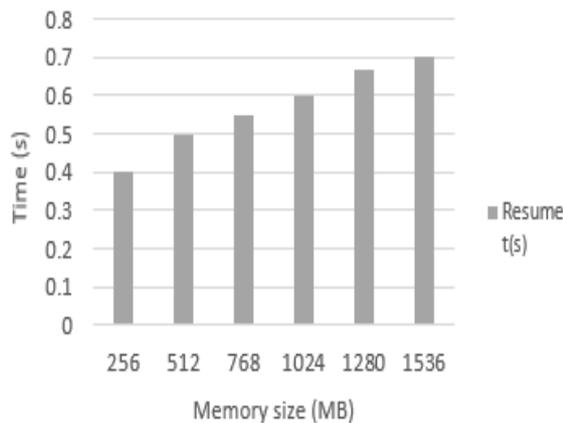


Fig.6. Plot of Resume phase

Also, the downtime of the VM as shown below in Fig. 7 was same (10.8 seconds) as total migration time and the total migrating was dependent on the size of the memory of the VM. This was a main drawback of offline migration as downtime was very large when services were not available.

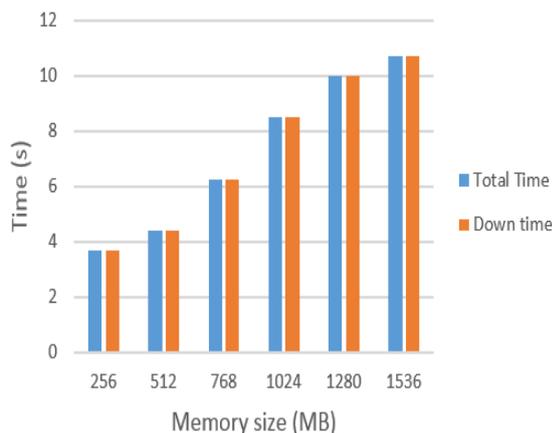


Fig.7. Plot of total migration time and downtime of VMs

B. Live Migration Analysis & Outputs

The major problem with offline migration was that it resulted in longer downtime. To reduce the same, live migration of the VM was performed where the VM moved from source host to the destination host in almost zero downtime. This technique of migration is preferred when high availability of services is required. Following command is used for live migration from command line-

```
# ssh root@10.16.171.23 virsh -connect
qemu+ssh://root@10.16.171.9 migrate --live 10.16.171.23
qemu+ssh://10.16.171.23/system
tcp://10.16.171.23:48160
```

For live migration, pre-copy approach of memory migration was used [6]. In the Pre-copy approach series of iterations, as shown in Fig. 8, were involved in transferring the memory. When migration was at the start only, dirty pages kept copying. When numbers of dirty pages were under threshold level, the source VM would stop and copy all remaining dirty pages. After this, the VM at last destination would be resumed [17]. This technique could take longer downtime, which depended on the writable set. Fig. 9 and 10 shows the plot of total migration time(12.5 seconds) and downtime(7 seconds) respectively. In live migration also, the total migration time depends on the size of memory. Downtime may or may not increase with memory size as it depends on the rate at which memory pages become dirty.

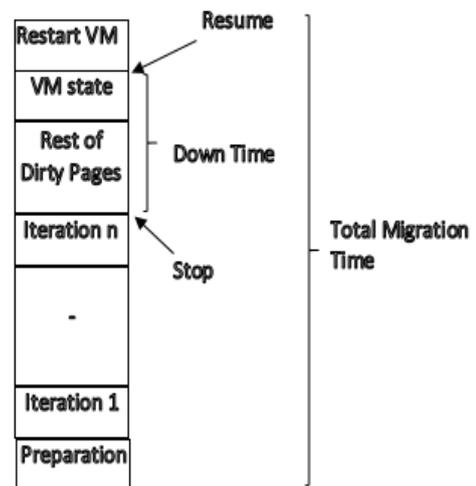


Fig.8. Pre-copy memory migration

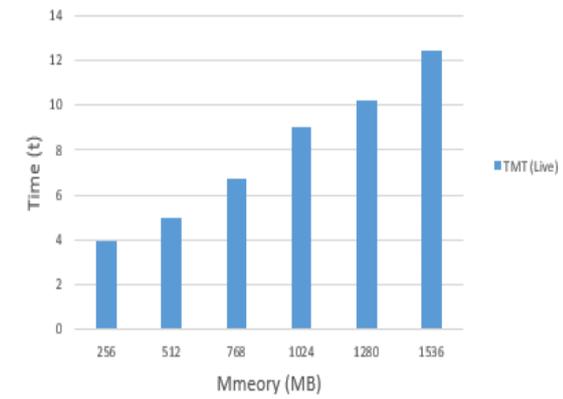


Fig.9. Plot for Total migration time

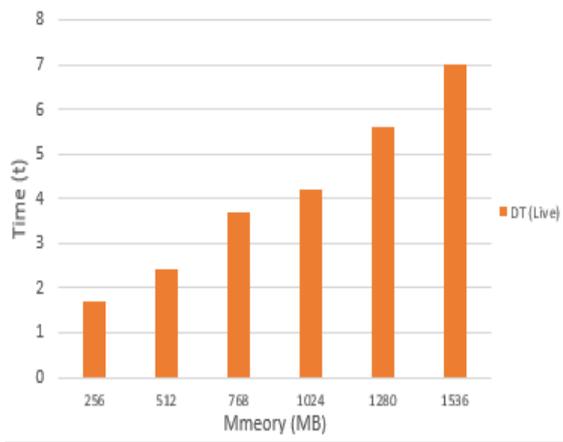


Fig.10. Plot for VM downtime

C. Performance Analysis of Offline and Live VM Migration

The comparison between the downtime in offline method and live migration, shown in Fig.11, made it clear that the downtime of offline migration was more (10.7seconds) in comparison to the live migration (7seconds).

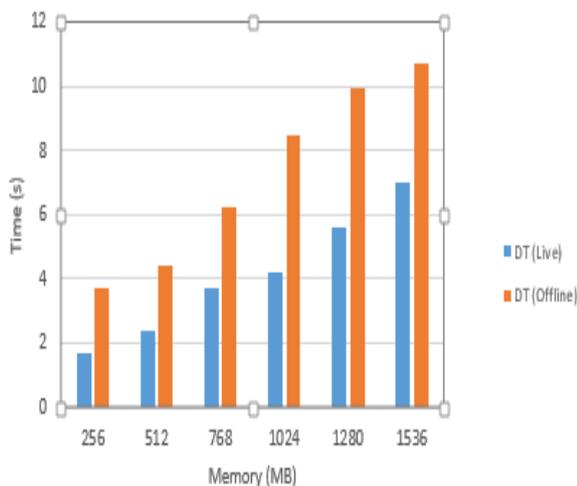


Fig.11. Plot of downtime offline vs. live

Further, the total migration time in live migration took more time(12.3 seconds) in comparison to the offline migrations(10.7 seconds) as shown in Fig. 12 below.

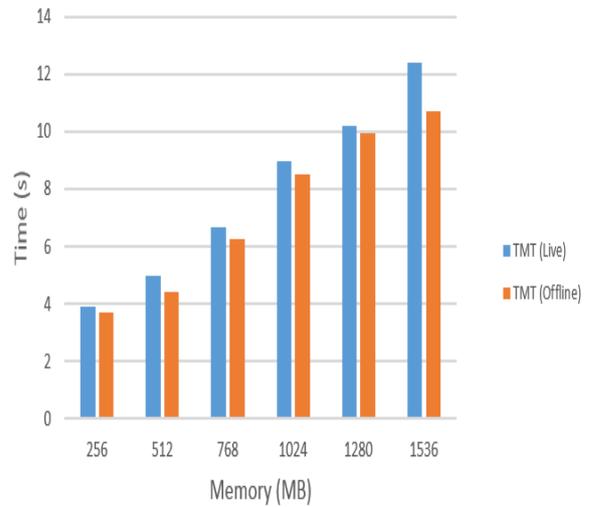


Fig.12. Plot of Total migration time offline vs. live

V. CONCLUSIONS

Components of different applications can be hosted with the creation of Virtual Machines which provides scalability and portability. Requirements on the cloud environment are dynamic; therefore, there is always a need to move virtual machines within the same cloud or between different clouds. There can be various reasons for VM migrations like saving energy of the host which is less in use, handling fault tolerance if some host is not working properly and balancing the load among all hosts.

In this paper, we performed experiments to do the performance analysis of VM migration using KVM virtualization. VM migration, offline as well as live, was performed to analyze two important parameters viz. total downtime and total migration time. After comparison, the total migration time and downtime in both cases (offline and live) it was found that the downtime during live migration was very less in comparison to the offline migration while the total migration time was more in comparison to the offline migration.

From the experiment, it can be concluded live migration is a good choice if VM is running an important application/s which should not be down for a longer time. However, if the requirement is to only migrate the VMs in less time, offline migration will be a better option.

The current study was conducted using open source hypervisor KVM as it is a less expensive option to a proprietary hypervisor such as Xen and VMWare. The future work may be conducted to show a comparison of performance analysis between Xen and VMWare hypervisor with KVM so as to throw more light on the efficient choice of hypervisors for Virtualization. The study can further be expanded by including a comparison

of performance analysis by running more heavy versus light applications on VMs while migration. d of the paper.

ACKNOWLEDGMENT

We want to offer our sincere thanks to all the members of Computer Service Center of IIT Delhi and especially to Dr Gopal Krishan for his extraordinary support and guidance. Also, we would like to thank Dr Hemraj Verma for his valuable suggestions.

REFERENCES

- [1] R. Buyya, C. Vecchiola, S. T. Selvi. "Mastering Cloud Computing: Cloud Foundation and Applications Programming", MK Publisher Edition 2013.
- [2] M. Imran Alam, M. Pandey, S. S. Rautaray, "A Comprehensive survey on Cloud Computing", International Journal of Information Technology and Computer Science, 02, pp. 68-79, 2015.
- [3] M. Durairaj, P. Kannan, "A Study On Virtualization Techniques And challenges in cloud computing", International Journal of Scientific and Technology Research, Vol-3, issue-11, pp 147-151, Nov 2014.
- [4] N. el-Khameesy, H.A. Rahman, "A Proposed virtualization technique to enhance IT Services", International Journal of Information Technology and Computer Science, 12, pp. 21-30, 2012.
- [5] A. Rabiatal Addawiyah Mat Razali, A. Ruhani, Z. Norliza, S. Mustafa, "Virtual Machine Migration Implementation in Load Balancing for Cloud Computing", proceedings of IEEE conference Intelligent and Advanced Systems, (Kuala Lumpur), pp. 1-4, 2014.
- [6] P. Getzi Jeba Leelipushpam, J. Sharmila, "Live VM Migration Techniques in Cloud Environment- A survey", Proceedings of IEEE Conference on Information and Communication Technologies, (Je Ju Island), pp. 408-413, 2013.
- [7] F. Salfner, P. Tröger, A. Polze, "Downtime Analysis of Virtual Machine Live Migration", The Fourth International Conference on Dependability, (IARIA), pp. 100-105, 2011.
- [8] W. Hu, A. Hicks, L. Zhang, E. M. Dow, V. Soni, H. Jiang, R. Bull, J. N. Matthews, "A quantitative study of virtual machine live migration", Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference, (August 05-09, Miami, Florida, USA), 2013.
- [9] S. Akoush, R. Sohan, A. Rice, A. W. Moore, A. Hopper, "Predicting the Performance of Virtual Machine Migration", IEEE proceedings of Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS, Miami Beach, FL), pp. 37-46, 2010.
- [10] D. Kapil, E. S. Pilli, R. C. Joshi, "Live virtual machine migration techniques: Survey and research challenges", IEEE proceedings of Advance Computing Conference, (Ghaziabad, India), pp. 963-969, 2013.
- [11] M. R. Anala, M. Kashyap, G. Shobha, "Application performance analysis during live migration of virtual machines", IEEE proceedings of Advance Computing Conference (Ghaziabad, India), pp. 366-372, 2013.
- [12] M. Zhao, R. J. Figueiredo, "Experimental study of virtual machine migration in support of reservation of cluster resources", Proceedings of the 2nd international workshop on Virtualization technology in distributed computing, (Reno, Nevada), pp. 1-8, 2007.
- [13] H. R. Prakash, M. R. Anala, G. Shobha, "Performance analysis of Transport Protocol during live migration of Virtual Machines", Indian Journal of Computer Science and Engineering, Vol-2 no-5, pp. 715-722, 2011.
- [14] A. Shribman, B. Hudzia, "Pre-Copy and Post-Copy VM Live Migration for Memory Intensive Applications", Springer Lecture notes in Computer Sc., vol. 7640, pp. 539-547, 2013.
- [15] A. Gupta, J. Kumar, D. J Mathew, S. Bansal, S. Banerjee, H. Saran, "Design and Implementation of the Workflow of an Academic Cloud", Proceedings of the 7th International conference on databases in networked information systems Springer Lecture notes in computer science, pp. 16-25, 2011.
- [16] Y. Wu, M. Zhao, "Performance modeling of virtual machine live migration", Proceedings of IEEE 4th international conference on cloud computing, pp. 492-499, 2011.
- [17] X. Feng, J. Tang, X. Luo, Y. Jin, "A performance study of live VM migration technologies: Motion vs. XenMotion", Proceedings of IEEE conference Communications and Photonics, Shanghai, pp. 1-6, 2011.
- [18] Red Hat Inc. Kvm - kernel based virtual machine. Technical report, Red Hat Inc., 2009 <https://www.redhat.com/en/resources/kvm-%E2%80%93-kernel-based-virtual-machine>

Authors' Profiles



Garima Rastogi is a research scholar in DIT University Dehradun, Uttarakhand, (INDIA) attached to the Department of Computer Science and Engineering. She has completed M.Tech in Computer Science Engineering and pursuing a PhD in the field of Cloud Computing from DIT University, Dehradun. She has published 15 research papers in different reputed journals and national and international proceedings. She is a member of Computer Society of India. She has also qualified UGC Net in Computer Science.



Prof (Dr) Rama Sushil currently working as prof and head, Department of Information Technology, DIT University, Dehradun. She has completed PhD from IIT Roorkee, Uttarakhand, (INDIA). She has guided various PhD students. She has also authored chapters in the books of IGI Global and published more than 40 research papers. Her research interests are Cloud Computing, Mobile Agents and Distributed Computing.

How to cite this paper: Garima Rastogi, Rama Sushil, "Performance Analysis of Live and Offline VM Migration Using KVM", International Journal of Modern Education and Computer Science(IJMECS), Vol.8, No.11, pp.50-57, 2016.DOI: 10.5815/ijmeecs.2016.11.07