Journal of Engineering and Applied Sciences 12 (13): 3401-3406, 2017
ISSN: 1816-949X

Efficient Virtual Machine Migration Using Open Source Hypervisor

[1]Garima Rastogi and [2]Rama Sushil
[1]Department of Computer Science and Engineering, DIT University, Dehradun, India
[2]Department of Information Technology, DIT University, Dehradun, India

Abstract: To fulfil the dynamic requirements in cloud environment there is always a need to migrate virtual machines from one place to another. It also helps in managing load balancing, power saving and maintenance activities such as management of faults due to resource failures, power failure, etc. There are two types of hypervisors which can be used for virtualization, i.e., open source and proprietary hypervisor. Most of the time practitioners have used proprietary hypervisors as compared to open source hypervisors because they consider open source hypervisors have less functionality. This research study focuses on to examine the efficient virtual machine migration technique in various scenarios with the motive to save cost. The study uses Kernel-based virtual machine, open source hypervisor for the experiment. The study also compares the virtual machine migration results using open source hypervisor, KVM with a proprietary hypervisor, Xen.

Key words: Virtualization, virtual machine, migration, hypervisor, KVM

INTRODUCTION

Cloud computing helps organizations to reduce their infrastructural and computing cost by making sharing of resources possible. Through coherence and economies of scale, it enables organizations to achieve utility computing. The spirit of hospitality and tourist acceptance and foster and strengthen this scientific and necessary branch of science in society sing a specific behavior-scientific base.

One of the weaknesses of domestic tourism industry, cloud computing appears 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 it is more powerful in terms of functionalities to share and utilize resources more efficiently and effectively (Buyya *et al*., 2013; Kerr and Davari, 2013). National Institute of Standard Technology (NIST) has given five key features of cloud computing-rapid elasticity, resource pooling on demand self-service, extensive network access and fifth one is measured services. In this, resources are pooled at the centralized places known as data centers that are accessible from everywhere as per the demand (Kerr and Davari, 2013).

An important technology that is used for implementing cloud computing is virtualization. This technology has diverted the industry perspective from the utilization of resources physical to logical. It creates an abstract layer over the actual hardware and software (Kerr and Davari, 2013). It utilizes single physical machine in the form of 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, network, etc. (Rastogi and Sushil, 2015). There can be various types of virtualizations like:

Application virtualization: It is a technology that provides the virtual environment in which the application can be executed without having install on the computer. The application behaves same as the local application on the client system. For, e.g., VMW are Thinapp, Oracle secure Global desktop, App-V, etc.

Storage virtualization: This technology provides the pooling of multiple physical storage devices which are accessed by single domain name. Through this, distributed storage is managed in such a way as if it is one large 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 (Ahmad *et al*., 2015).

Server virtualization: This technology provides the masking of existing server with all its resources (Rastogi and Sushil, 2015). 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.

Corresponding Author: Garima Rastogi, Department of Computer Science and Engineering, DIT University, Dehradun, India

The main activity in virtualization is creation and migration of Virtual Machine (VM). By creating VMs, it collaborates multiple unutilized resources into a shared resource pool and utilizes them by performing different tasks simultaneously to fulfill multiple user demands. VMs can execute various tasks as per the requirements of clients. The resources can be allocated or de-allocated dynamically on VMs which converts single physical host into number of virtual hosts (Masdari *et al*., 2016). Generally, VM migration are of two types-one is live migration and another is offline migration. In live migration a running VM is migrated from one host to another. The goal of live migration is to minimize the interruption of services that are running on a VM during migration (Clark *et al*., 2005; Refaat *et al*., 2016). In offline migration, first a VM is suspended, then all files related to the configuration and VM memory image is moved from source to destination host. At the end of this migration, the copied VM image is resumed at the destination host Most of the vendors of virtualization technology like Xen, Kernel-Based virtual Machine (KVM) and Hyper-V, etc., used live migration as an important feature as it contributes significantly for their sales. However, not all live migration technologies are equal in all aspects. One technology may focus on minimizing the downtime of VM migration while other may emphasize on minimizing the total migration time (El-Khameesy and Mohamed, 2012; Baruchi *et al*., 2015).

The current study examines the efficient VM migration technique in some scenarios by using an experiment on KVM and Xen hypervisors. The main motive of the study is to find which migration technique is better in a given situation with lesser cost. To find out the main foal of the study some experiments have been conducted on open source hypervisor (KVM) as well as proprietary hypervisor (Xen) and the performance of both hypervisors is then compared. The parameters that are considered for analysis are Total Migration Time (TMT), Down Time (DT), Migration Data (MD) and Migration Time (MT).

MATERIALS AND METHODS

Preliminaries and methods: This study is based on the experimental analysis. Two physical hosts of same configuration are used for the experiment. The configuration of the hosts used is 2×4 core Intel (R) Xeon (R) CPU E5540 of 2.53 GHz with 12 GB RAM. Shared storage has been used of 50 TB which can be accessed by both of the host servers. The whole experiment is repeated for KVM and Xen one by one. Figure 1 shows VM setup in host-1.

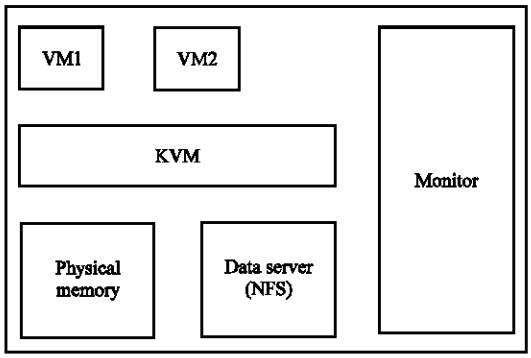

Fig. 1: Virtual machine setup in Host-1

The performance of a live migration is dependent on two parameters CPU state and memory state. While VM migration, VM's CPU state is needed to be switched from source host to destination host. It consumes some amount of time to transfer the information. Similarly, memory state of VM also needs to be transferred. But is quite large amount in comparison to CPU state because it includes state of Operating System (OS) and all processes running in the VM (Hu *et al*., 2013).

Categories of memory contents to be migrated: There are five categories of memories which play an important role in VM migration. All have a relationship amongst each other with respect to the size (Akoush *et al*., 2010). First category of memory is called as configured memory to VM which is an amount of memory given to the VM by a hypervisor. It is also called as physical memory available for use. Second category, called allocated memory is an amount of physical memory which the hypervisor has actually allocated to VM. It is always less than the configured memory that is being used by VM. Third category of memory is used memory. It is a memory which is used by a VM OS. These are memory pages that reside inside VM memory. Fourth category is called as Request Memory by Application and is an amount of memory required by applications that are running inside VM. Last category, called as Dirtied Memory is a part of requested memory of an application that is actively modifying via writing in-memory pages (Anala *et al*., 2013; Liu and He, 2014; Shribman and Hudzia, 2012; Kim *et al*., 2011). For live migration, configured memory can be the upper bound to estimate the migration time. Dirty memory is also an important parameter which can increase the total amount of data to be transferred (Isci *et al*., 2011).

Performance metrics: This experiment has measured some important performance metrics of live migration such as TMT, DT and data transferred over the network during migration. Ping test has been applied to check the accessibility of VM while migration.

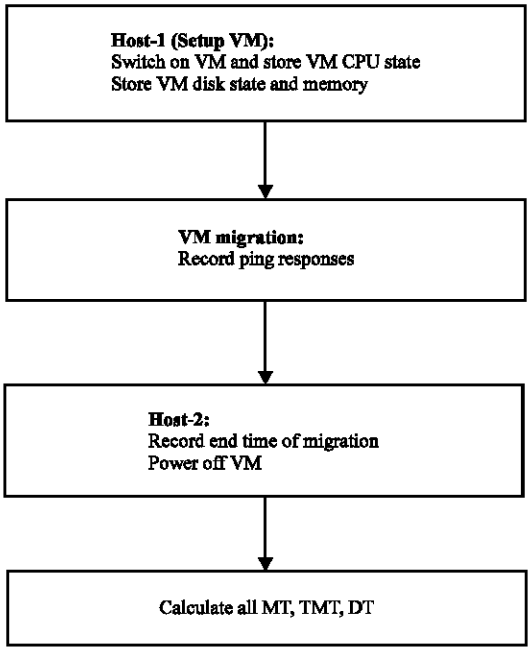

Fig. 2: Steps to evaluate performance metrices

The ping test helps in capturing time stamps and exact pattern of live migration (El-Khameesy and Mohamed, 2012; Feng *et al*., 2011). TMT is measured by noting down the time at the start as well as at the end of the migration. Similarly, to measure the DT of VM, ping test is used during VM migration. Each time, timestamp and sequence number are noted down where no response is received from VM. For doing above parametric measurements, following shell script is used in the experiment. The steps are shown in Fig. 2.

RESULTS

In the experiment, VM live migration is done from host 1-2.

Senario-1: Used memory for VM is approximately 125 and 90 MB for KVM and Xen, respectively. Migration data is 258 and 2300 MB for KVM and Xen respectively. The TMT and DT are recorded through ping test. The migration time is also calculated by deducting the DT from the TMT. All the results are plotted using bar charts in Fig. 3-5. It is very clear from the figures that KVM took less time in comparison to Xen in all cases.

The data transfer speed is calculated by using migration time and migration data. It is approximately 19 and 79 MB/s for KVM and Xen respectively. This shows that Xen has better throughput in comparison to KVM. It is clear from the plot shown in Fig. 4 that KVM takes less DT because it can synchronize dirty memory

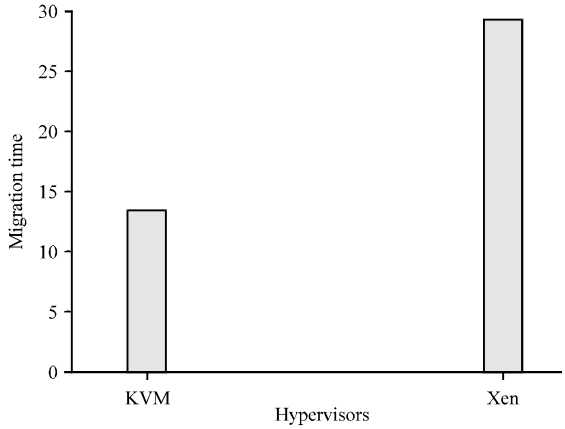

Fig. 3: Plot of migration time vs. hypervisors

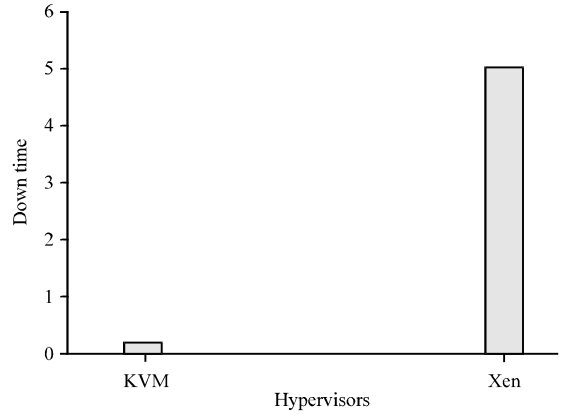

Fig. 4: Plot of downtime vs. hypervisors

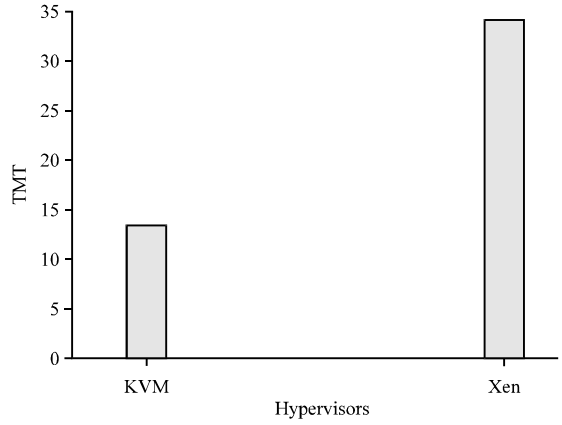

Fig. 5: Plot of total migration time vs. hypervisors

data fast to achieve less DT. The main reason for this is KVM transfers only allocated memory but Xen migrates whole configured memory even when the actual usage is less.

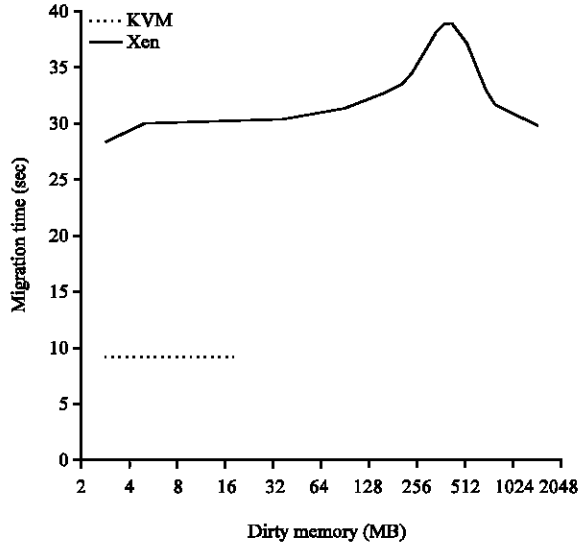

Fig. 6: Plot of dirty memory size vs. migration time for all virtualizations

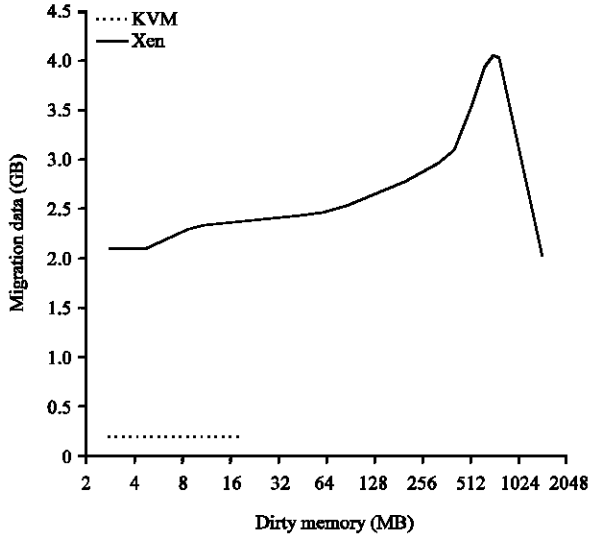

Fig. 7: Plot of dirty memory size vs. migration data for all virtualizations

Senario-2: To explore the impact of dirty memory, its size is increased gradually in migrated VM. The responses of migration time, data and DT are shown in Fig. 6-8, respectively with respect to increasing dirty memory size. The most important finding is that KVM fails to finish migration once dirty memory size reaches to 32 MB with the current setup. It stops responding, i.e., it shows no progress. Further, once dirty memory size reaches to the configured memory size, the migration time, down time and migration data for Xen decreases instead of increasing (Chowdhury and Boutaba, 2010).

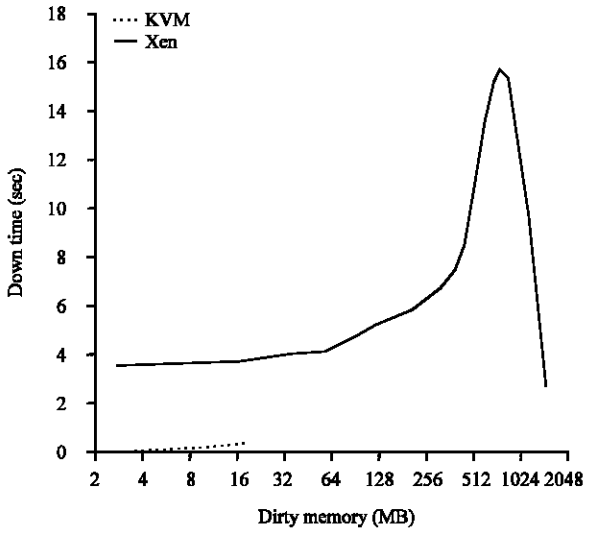

Fig. 8: Plot of dirty memory size vs. down time for all virtualizations

From all above analysis it is concluded that live migration in KVM is good if the dirty memory size is less. But if the dirty memory size is large or continuously increasing with applications running inside VM, then it is better to prefer offline VM migration (Jia *et al*., 2015). Same thing is also true for Xen as the impact of dirty memory size on DT is quite significant. It takes long time of approximately 16-17 sec for migration.

DISCUSSION

Several research studies have been done related to the VM migration to analyze the performance of VM migration by using different hypervisors. In a study done to evaluate the effects of live VM migration using Xen hypervisor tested the performance of VM by running modern internet application on it in Service Level Agreement (SLA) oriented environment. The result shown in the study is acceptable but this whole study was done by using Xen, a proprietary hypervisor, only. Its performance is not compared using any other hypervisor. Researchers have also done study on the techniques of VM migration in vehicular clouds (Refaat *et al*., 2016). They have shown the comparative study between various schemes and found best performing scheme. Researchers have done simulation using MATLAB software. In another study researchers focused on the performance of VM live migration by considering workload as an important factor which can have great impact the performance of live VM migration. They have also shown

the comparison with traditional techniques (Baruchi *et al*., 2015). Hu *et al*. (2013), researchers performed experimental analysis to analyze the performance of VM live migration with respect to downtime and total migration time. They have used two hypervisors VMW are and Xen for this analysis. The study concluded that the performance of live VM migration can vary according to the memory load. In another study, Akoush *et al*. (2010) have done an experiment on VM live migration which measures important characteristics like TMT and unresponsive time of VM. They have done experiment using KVM, Xen and VMW are. They also suggested some parameters in the study to improve the performance of live migration. In a coordination system, proposed for VM migration in cloud, Prakash tried to minimize the migration cost by reducing total migration time and downtime.

The current study is unique in the sense that it identifies the efficient VM migration technique using open source hypervisor with the motive of saving cost. The study may benefit cloud service providers that are always looking for ways to reduce costs. From the experiment, it can be seen that the KVM, an open source hypervisor is a good option while implementing virtualization, especially for organizations that are concerned about cost. KVM is a less expensive option to a proprietary hypervisor such as Xen.

CONCLUSION

One of main purpose of this study was to find out if live migration could be performed efficiently using open source hypervisors or not. A series of experiments were done using open source hypervisor (KVM) as well as proprietary hypervisor (Xen) and the performance of both hypervisors was compared on parameters like Total Migration Time (TMT), Down Time (DT), Migration Data (MD) and Migration Time (MT). Results of these experiment showed that migration data and DT were very less using KVM in comparison to Xen. Dirty memory size had significant impact on migration in KVM because KVM was unable to finish migration once the dirty memory size reached to 32 MB in the proposed setup. In case of Xen, it took approximately 16-17 sec which was also a large duration. It can be concluded that live migration in KVM is good if the dirty memory size is less in comparison to Xen. But if the size of dirty memory is increasing then offline migration in KVM should be preferred. In future this study can be enhanced by comparing performance of KVM with other hypervisors by taking some more scenarios and configuration setup.

REFERENCES

Ahmad, R.W., A. Gani, S.H.A. Hamid, M. Shiraz and A. Yousafzai *et al*., 2015. A survey on virtual machine migration and server consolidation frameworks for cloud data centers. J. Network Comput. Appl., 52: 11-25.

Akoush, S., R. Sohan, A. Rice, A.W. Moore and A. Hopper, 2010. Predicting the performance of virtual machine migration. Proceedings of the 2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, August 17-19, 2010, IEEE, Cambridge, UK. isBN:978-1-4244-8181-1, pp: 37-46.

Anala, M.R., M. Kashyap and G. Shobha, 2013. Application performance analysis during live migration of virtual machines. Proceedings of the 2013 IEEE 3rd International Conference on Advance Computing (IACC), February 22-23, 2013, IEEE, Bangalore, India isBN:978-1-4673-4527-9, pp: 366-372.

Baruchi, A., E.T. Midorikawa and L.M. Sato, 2015. Reducing virtual machine live migration overhead via workload analysis. IEEE. Latin Am. Trans., 13: 1178-1186.

Buyya, R., C. Vecchiola and S.T. Selvi, 2013. Mastering Cloud Computing: Cloud Foundation and Applications Programming. 1st Edn., McGraw Hill Education, India,.

Chowdhury, N.M.M.K. and R. Boutaba, 2010. A survey of network virtualization. Comput. Networks, 54: 862-876.

Clark, C., K. Fraser, S. Hand, J.G. Hansen and E. Jul *et al*., 2005. Live migration of virtual machines. Proceedings of the 2nd Conference on Networked Systems Design & Implementation Vol. 2, May 02-04, 2005, USENIX Association, Berkeley, California, pp: 273-286.

El-Khameesy, N. and H.A.R. Mohamed, 2012. A proposed virtualization technique to enhance IT services. Intl. J. Inf. Technol. Comput. Sci., 4: 21-30.

Feng, X., J. Tang, X. Luo and Y. Jin, 2011. A performance study of live VM migration technologies: VMotion vs XenMotion. Proceedings of the Asia Conference and Exhibition on Communications and Photonics, November 13-16, 2011, Optical Society of America, Shanghai, China isBN:978-0-8194-8958-6, pp: 83101B-83101B.

Hu, W., A. Hicks, L. Zhang, E.M. Dow and V. Soni *et al*., 2013. A quantitative study of virtual machine live migration. Proceedings of the 2013 ACM Conference on Cloud and Autonomic Computing, August 05-09, 2013, ACM, Miami, Florida, USA. isBN:978-1-4503-2172-3, pp: 11-11.

Isci, C., J. Liu, B. Abali, J.O. Kephart and J. Kouloheris, 2011. Improving server utilization using fast virtual machine migration. IBM. J. Res. Dev., 55: 4-11.

Jia, X., J. Wang, C Huang, Q. Liu and K. He *et al*., 2015. Dynamic resource allocation based on energy utility maximization using virtual machines in cloud environment. Comput. Syst. Sci. Eng., 30: 1-10.

Kerr, C. and S. Davari, 2013. Cloud computing: Business trends and the challenges. Intl. J. Recent Trends Eng. Technol., 8: 76-83.

Kim, K.H., A. Beloglazov and R. Buyya, 2011. Power-aware provisioning of virtual machines for real-time cloud services. Concurrency Comput. Pract. Experience, 23: 1491-1505.

Liu, H. and B. He, 2015. VMbuddies: Coordinating live migration of multi-tier applications in cloud environments. IEEE. Trans. Parall. Distrib. Syst., 26: 1192-1205.

Masdari, M., S.S. Nabavi and V. Ahmadi, 2016. An overview of virtual machine placement schemes in cloud computing. J. Network Comput. Appl., 66: 106-127.

Rastogi, G. and R. Sushil, 2015. Cloud computing implementation: Key issues and solutions. Proceeings of the 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), March 11-13, 2015, IEEE, Dehradun, India isBN:978-9-3805-4415-1, pp: 320-324.

Refaat, T.K., B. Kantarci and H.T. Mouftah, 2016. Virtual machine migration and management for vehicular clouds. Veh. Commun., 4: 47-56.

Shribman, A. and B. Hudzia, 2012. Pre-Copy and Post-Copy VM Live Migration for Memory Intensive Applications. In: Euro-Par 2012: Parallel Processing Workshops, Loannis, C., M. Alexander, M.B. Rosa, M. Cannataro and C. Alxandru et al., Springer, Berlin, Germany isBN:978-3-642-36948-3, pp: 539-547.