International Journal of Soft Computing 11 (7): 453-458, 2016
ISSN: 1816-9503

Empirical Analysis of Live VM Migration Using KVM and Xen

[1]Garima Rastogi, [2]Rama Sushil and [3]Arun Kumar Yadav
[1]Department of Computer Science and Engineering,
[2]Department of Information Technology,
DIT University, 248009 Dehradun, India
[3]Department of Computer Science and Engineering,
Ajay Kumar Garg Engineering College, Ghaziabad, India

Abstract: Virtual Machine (VM) migration is an important activity which is required for managing virtualized environment, load balancing, power saving and maintenance activities like management of faults due to resource failures etc. For virtualization, practitioners have used mostly proprietary hypervisors as compared to open source hypervisors because they consider open source hypervisors have less functionality. However, some research studies indicate that open source hypervisors may be a good alternative without compromising on functionalities needed for virtualization. To confirm the same, an experiment has been carried out on two virtualization systems Kernel-Based Virtual Machine KVM (open source hypervisor) and Xen (proprietary hypervisor) to evaluate their respective performances total migration time, downtime etc. using VM live migration technique.

Key words: Virtualization, virtual machine, hypervisor, load balancing, live migration

INTRODUCTION

For implementing cloud computing, an important technology is used virtualization (Buyya *et al*., 2013; Kerr and Davari, 2013). This technology has diverted the industry perspective to utilize resources from physical to logical. The main goal of virtualization is to utilize the maximum capacity of available resources such as processor, storage, network etc (Rastogi and Sushil, 2015; Ahmad *et al*., 2015; Chowdhury and Boutaba, 2010). Through Virtualization, Virtual Machines (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 (Chowdhury and Boutaba, 2010; Masdari *et al*., 2016; Clark *et al*., 2005). VM migration can be of two types i.e., offline and live. 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. 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 (Refaat *et al*., 2016; El-Khameesy and Mohamed, 2012). Most of the vendors of virtualization technology like Xen, 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 (Baruchi *et al*., 2015; Leelipushpam and Sharmila 2013). The current study analyzes the performance of KVM and Xen hypervisors while carrying out VM live migration on the basis of some parameters like Total Migration Time (TMT), Down Time (DT), etc. Some important factors and categories of memory contents used in migration are The experimental setup is given in.

MATERIALS AND METHODS

Performance parameters of live migration: The performance of a live migration is depended on some important factors mentioned as below.

CPU state: When migration is done, a VM's CPU state is required to be switched from one (source) to other (destination) host. It takes small amount of time to transfer the information. This also contributes in the migration downtime (Sun *et al*., 2016).

Memory state: Memory state of VM also needs to be transferred from source to destination host. It is a quite large amount of information in comparison to the CPU state. It includes the state of guest Operating System (OS) and all the processes running within a VM (Salfner *et al*., 2011). In some cases, a VM is configured for more memory than the actual used memory. Here,

Corresponding Author: Garima Rastogi, Department of Computer Science and Engineering, DIT University, 248009 Dehradun, India

an efficient hypervisor can identify the unused memory and tries to transfer only contents of used memory and thus helps in reducing the migration time (Hu *et al*., 2013).

Categories of memory contents to be migrated: There are various categories of memories which play an important role in VM migration. All have a relationship among each other with respect to the size (Akoush *et al*., 2010).

Configured memory to VM: It is an amount of memory given to the VM by a hypervisor. It is also called as physical memory available for use.

Allocated memory: It 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.

Used memory: A memory which is used by a VM OS. There are memory pages that reside inside VM memory.

Request memory by application: Amount of memory required by applications that are running inside VM.

Dirtied memory: It 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, 2015; Prakash *et al*., 2011; Shribman and Hudzia, 2012; Kim *et al*., 2011). The relationship between all above memories is shown in (Fig. 1).

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 (Shribman and Hudzia, 2012).

Experiment setup: This experiment has been conducted in the Computer Service Center (CSC) lab of IIT Delhi. The resources of CSC data center of their private cloud Baadal have been used for the experiment. Two hosts were used for the experiment. The configuration of the host used was 2×4 core Intel (R) Xeon (R) CPU E5540 of 2.53 GHz and 12 GB RAM. Both host servers could access the shared storage which was of 50TB based on NetApp 3210V NAS and HP EVA6400 SAN with FC disks. For virtualization KVM (Kernel Virtual Machine) and Xen Hypervisors were used.

The experimental setup which shown in the Fig. 2. Two hosts of same configuration named Host-1 and Host-2 were taken. Host-1 had a number of VMs on it with OS Ubuntu 12.04 (Isci *et al*., 2011). 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

Table 1: Image format of virtual machines

Hypervisor	Memory migration image	Storage migration image
KVM	QCOW2	QCOW2
Xen	VHD	LVM

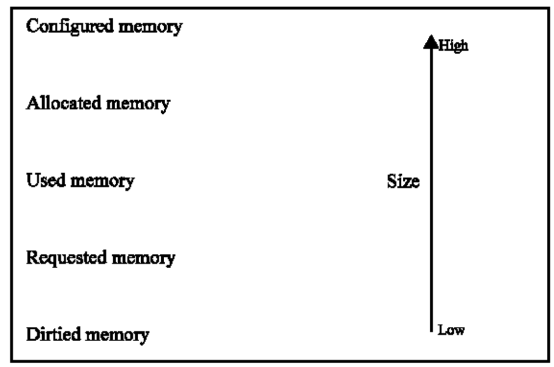

Fig. 1: Hierarchy of memories according to the size

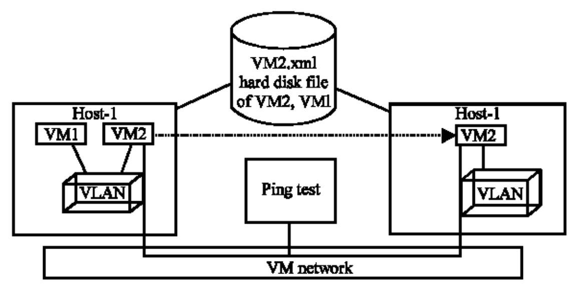

Fig. 2: Experimental setup

for VM migration between hosts; Secured Shell (SSH) protocol was used to provide secure data exchange between hosts on the network. Though there were various protocols available for data exchange for examples File Transfer Protocol (FTP), Post Office Protocol (POP) and Telnet but these were not secure as one could transfer information in the form of plain text, making it easy for hackers to access it (Feng *et al*., 2011; Masdari *et al*., 2016). 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. While creation of VM, image format of VM was different for both hypervisors as shown in Table 1.

This experiment measured an important performance metrics of live migration such as TMT, DT and data transferred over the network during migration. The experiment applied ping test to check the accessibility of VM while migration. The ping test helped in capturing time stamps and exact pattern of live migration.

Steps for evaluation of live migration: Following metrics were collected while migration of VM (Kim *et al*., 2011):

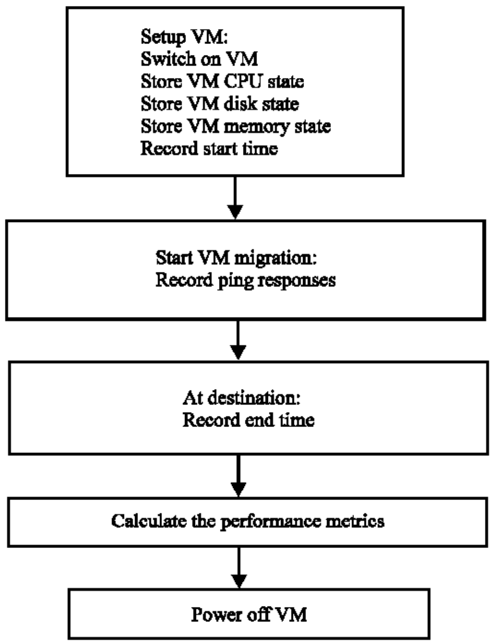

Fig. 3: Steps to evaluate performance metrics

- Total time taken by VM from source to destination i.e. Total Migration Time (TMT)
- Total time when VM is unresponsive while migration, i.e., Down Time (DT)
- Amount of data that is transferred over the network, i.e., migrated data

To measure the first parameter TMT, time was noted at the start as well as at the end of the migration. Similarly, to measure the DT of VM, ping test was used during VM migration. Each time, timestamp and sequence number were noted down where no response was received from VM. For doing above measurement, shell script was used in the experiment. The steps of script are shown in Fig. 3.

RESULTS AND DISCUSSION

In the experiment, VM live migration was done from host-1 to host-2. For KVM, used memory for VM is approximately 125 MB and migration data is 258 MB. For Xen, used memory is 90 MB and migrated data is 2300 MB. The TMT and DT were recorded through ping test. The migration time is also calculated by deducting the DT from the TMT shown in Table 2.

The above results have been shown using bar charts in the figures given below. Types of hypervisors are mentioned on x-axis. Migration time, DT and TMT are

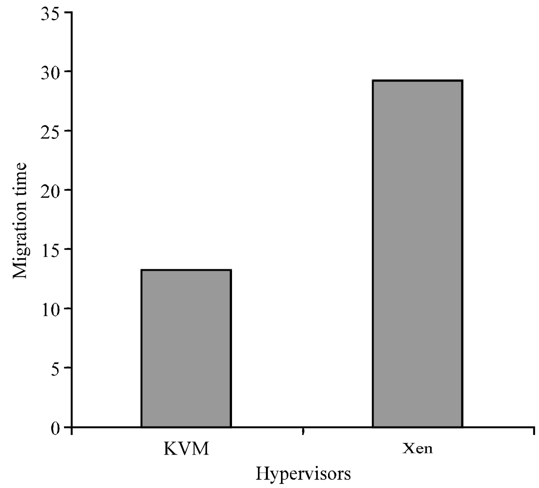

Fig. 4: Plot of migration time with respect to virtualizations (Hypervisors)

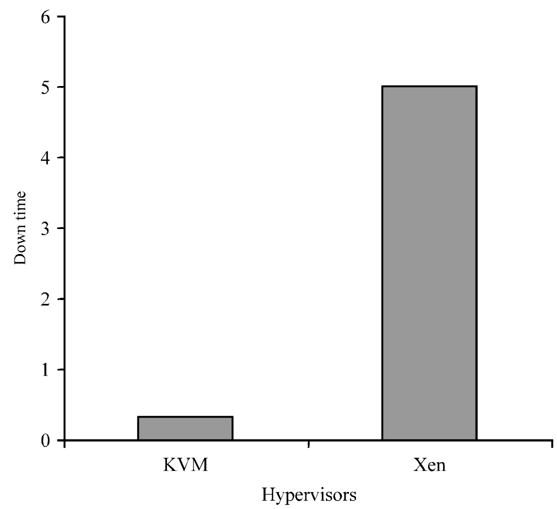

Fig. 5: Plot of virtualizations (Hypervisors) vs. downtime

shown on y-axis in Figs. 4-6 respectively. It is apparent that KVM took less time in comparison to Xen in all cases. The data transfer speed was calculated by using migration time and migration data. It was approximately 19 $MBsec^{-1}$ and 79 MB sec^{-1} for KVM and Xen respectively. This showed that Xen had better throughput in comparison to KVM. It was clear from the plot that KVM took less DT, i.e., it can synchronize dirty memory data fast to achieve less DT. The main reason for this is that KVM transferred only allocated memory but Xen migrated whole configured memory even when the actual usage was less. Dirty memory size has g reat impact on the performance of live migration (Baruchi *et al*., 2015). To explore the impact of dirty memory, size of dirty memory is increased

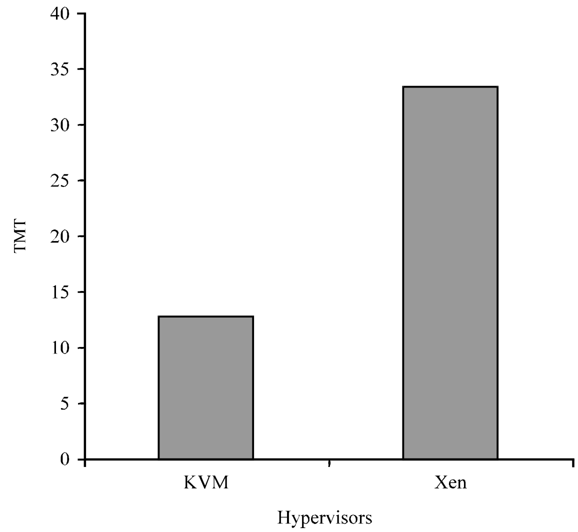

Fig. 6: Plot of virtualizations (Hypervisors) vs. total migration time

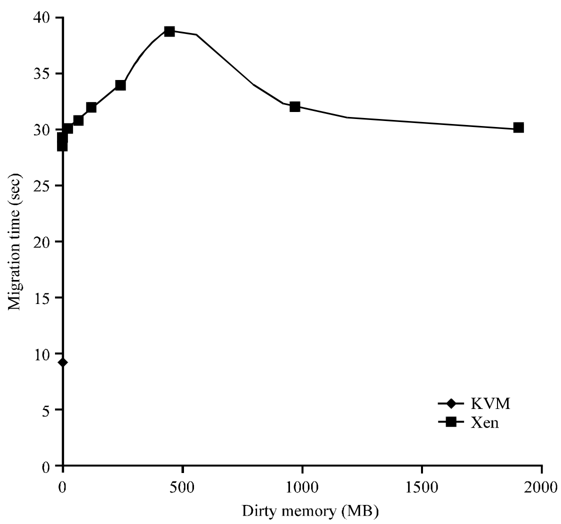

Fig. 7: Plot of dirty memory size vs. migration time for all virtualizations

Table 2: Performance metrics for KVM and Xen

Hypervisor time (sec)	Migration time (sec)	Down time (sec)	Total migration time (sec)
KVM	13.35	0.16	13.41
Xen	29.19	5.01	34.20

gradually in migrated VM. The responses of migration time, data and DT shown with respect to increasing dirty memory size is shown in Fig. 7-9. The most important result for KVM was that KVM failed to finish migration once dirty memory size reached to 32 MB with given setup. It stopped responding, i.e., it showed no progress. Further, once dirty memory size reached to the configured

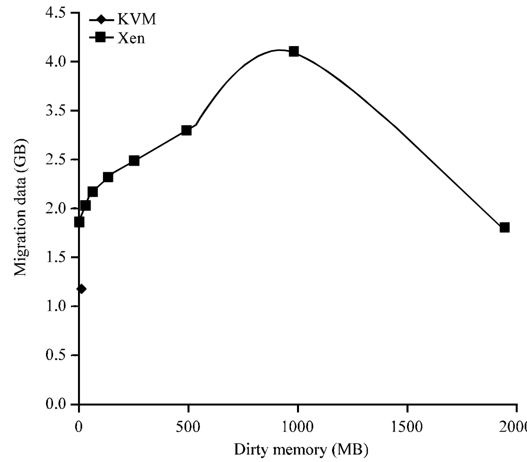

Fig. 8: Plot of dirty memory size vs. migration data for all virtualizations

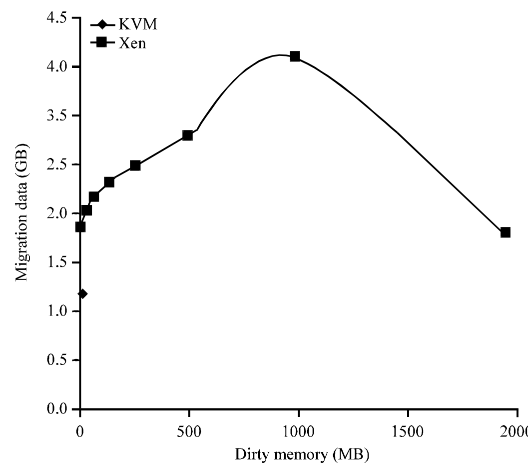

Fig. 9: Plot of dirty memory size vs. down time for all virtualizations

memory size, the migration time, down time and migration data for Xen decreased instead of increasing. 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. 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.

CONCLUSION

In this study, some important parameters viz. migration time, DT, TMT and migration data is analyzed for live VM migration. It is found that migration data and

DT is very less using KVM in comparison to Xen. Dirty memory size has great impact on migration in KVM because KVM is unable to finish migration once the dirty memory size is reached to 32 MB in the proposed setup. In case of Xen, it took approximately 16-17 sec which is also a large duration. It can be concluded from the results obtained from the experiment 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.

ACKNOWLEDGEMENTS

Researchers would like to acknowledge to all the members of Computer Service Center of IIT Delhi, especially Dr Gopal Krishan for his extraordinary support and guidance. The authors also thank to Dr Santosh Kumar, DIT University, Dr Hemraj Verma, DIT University for their constructive suggestions.

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. In: Cloud Foundation and Applications Programming, Buyya, R., C. Vecchiola and S.T. Selvi (Eds.). McGraw Hill Education, India, pp: 71-109.

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 Symposium on Networked Systems Design and Implementation, Volume 2, May 2-4, 2005, Boston, MA., USA., 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.

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.

Leelipushpam, P.G.J. and J. Sharmila, 2013. Live VM migration techniques in cloud environment-a survey. Proceedings of the 2013 IEEE Conference on Information and Communication Technologies (ICT), April 11-12, 2013, IEEE, Coimbatore, India, SBN:978-1-4673-5759-3, pp: 408-413.

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.

Prakash, H.R., M.R. Anala and D.G. Shobha, 2011. Performance analysis of transport protocol during live migration of virtual machines. Indian J. Comput. Sci. Eng., 2: 715-722.

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.

Salfner, F., P. Troger and A. Polze, 2011. Downtime analysis of virtual machine live migration. Proceedings of the Fourth International Conference on Dependability (DEPEND'11), August 21-27, 2011, IARIA, Berlin, Germany, pp. 100-105.

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.

Sun, G., D. Liao, V. Anand, D. Zhao and H. Yu, 2016. A new technique for efficient live migration of multiple virtual machines. Future Gener. Comput. Syst., 55: 74-86.