ISSN No. 0976-5697

Volume 5, No. 5, May-June 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Comparative Study of Pre-Copy based Live Virtual Machine Migration Techniques

Priyansh Srivastava[1], Surabhi Mour[2], Payal Patel[3], Hari Ram[4], Bhavesh N. Gohil[5]
[1,2,3,4,5]Department of Computer Engineering, SVNIT Surat, India

Abstract: Virtual Machine(VM) migration is a powerful management technique that gives data center users the ability to adapt the placement of VMs in order to better satisfy performance objectives, improve resource utilization and communication locality, load balancing, mitigate performance hotspots, achieve fault tolerance, reduce energy consumption, and facilitate system maintenance activities. Live migration defines movement of VMs form one physical host to another without disrupting the client or application. One of the objectives of live migration is that it should have minimum migration time as well as downtime so that applications running on VM are suspended for negligible time. In this paper we focus on various pre-copy based live virtual machine migration techniques and compare them based on key performance metrics like downtime, total migration time, total no of bytes transferred, etc.

Keywords: Virtualization; Virtual Machine; Live Migration; Downtime; Total Migration Time; Pre Copy

I. INTRODUCTION

The term "virtualization" came in 1960 mainfarmes as mainframes are logically divided for using them for different applications[1] . Serveral types of virtulization are used in IT technology now a days; main types being hardware virtlization, desktop virtualization, software virtualization and nested virtulization. Software virtualization has sub parts indicating different types of virtulization. In cloud computing, storage, application, server and network devices can be virtualized [2].

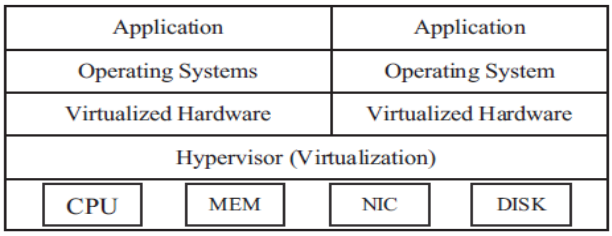

Figure. 1 – Virtualization [2]

Virtualization is a key concept of cloud computing [3]. Virtualization is vital to cloud computing because it provides the platform for optimizing complex resources which simplifies delivery of services along with maintaining scalability making cloud computing technology more effective. It can be applied very broadly to memory, networks, data, storage, hardware, operating systems and applications.

The main benefits of virtualization are given as below. Hardware Abstraction, Ease of Migration, Encapsulation of storage, Snapshots, Ease of Archiving, Ease of Growth, Improved monitoring and Troubleshooting, Workload Consolidation, Ease of environment Segmentation, Improved remote management.[4] As it allows different environments on a single host, the main issue with virtualization is the complexicity of licensing [1].

A Hypervisor is used to manage different aspects of virtualization. It allows multiple operating systems on a single hardware host fulfilling the requirement of cloud computing of having multiple operating environments on a single machine. XeN, QEMU and KVM are different hypervisors available.

The need for VM migration is: proactive maintenance- In case of imminent failure, the problem can be resolved before disruption of service occurs. Load balancing- balance load of the system to optimize available CPU resources. Fault tolerance - server consolidation, servers can be selectively brought down for maintenance after migrating their workload to other servers.

In this paper we present and compare different Live Virtual Machine Migration techniques based on various performance metrics and discuss its pros and cons. Section II describes brief introduction of Virtual Machine Migration, section III contains different techniques of Live VM Migration. Section IV refers to comparison of different techniques; conclusion of work goes in section V.

II. BACKGROUND AND LITERATURE SURVEY

Virtualization technology provides the feature of multiple OS running on the same physical machine. Virtualization provides the facility to migrate virtual machine from one host to another physical host. [2] Types of hardware virtualization are described below [1].

a. ***Full virtualization*:** Complete simulation of actual hardware to allow software.
b. ***Partial Virtualization*:** Some but not all of target environment is simulated.
c. ***Para Virtualization*:** Hardware is not simulated. The guest program has its own domain.

Virtual machine migration is a useful tool for an administrator of a data center and a cluster. It forms a clean layer between hardware and software. Process level migration problems can be avoided by migrating a VM. Virtual machine migration enables energy saving, load balancing and efficient resources utilization [2].

Virtual Machine Migration methods are divided into two parts: Hot (live) migration and Cold (non-live) migration. Types of VMs are based on hypervisor used by the particular system. In hot (live) migration the transfer of VM and its state and applications happen from one physical host to another without stopping the OS operations or applications. While cold (non-live) migration suspends the

VM, copies VM state and its applications to a different directory on the physical host and then migrates the VM. So clearly, the downtime is more than live migration. Cold migration easily avoids downtime caused by failure and maintenance issues with physical machines. Live migration facilitates the following benefits.

Load balancing: Dynamically balances workloads along physical machines to optimize overall system performance.

Proactive Maintenance: Frees up a given physical machine for maintenance without downtime for users.

Live migration can be done on memory, storage, network connectivity etc. [5]. But in this paper we focus memory migration of VMs. Memory migration is an integral part of VM migration, moving memory instance of VM from one host to another can be approached in number of ways. Virtual Machine downtime and total migration time defines two approaches of memory migration- Pre-Copy and Post-Copy. Pre copy approach is implemented in two phases [6].

a) Warm up phase: copy memory, implemented before VM reallocation.
b) Stop and copy phase: memory is copied after VM reallocation.

Pre copy challenge: It is difficult to apply test memory dirtying application as downtime will be increased fast.

Post copy challenge: Total migration time is more for fast memory reading. Application page fault is more.

III. PRE COPY BASED LIVE VIRTUAL MIGRATION TECHNIQUES

Other than traditional Pre-Copy approach, different techniques are introduced which are modifications of the naïve Pre-Copy approach and these techniques provide solutions to challenges formed by the traditional Pre-Copy approach.

Improved Pre-Copy approach [7] - System keeps the records of frequently modified pages. Three types of bitmaps – to_send, to_skip and to_fix are used to define modified pages. This approach reduces the number of iterations and total data transferred. System downtime is increased but total migration time is reduced.

This approach provides restrictions to avoid unnecessary transfer of dirty pages in iterations. Algorithm works in both high and low dirty pages situation. VMs modify their pages time to time. So the pages which are continuously modified are sent to the destination in the last phase using stop and copy approach. The modifying condition is determined by two bitmaps, to_send and to_skip whose values are set by the algorithm proposed in [8].

Recovery and CPU scheduling [9] – this approach provides fast and transparent live migration. In order to synchronize states of the source and target hosts, the target host executes the log files which are previously generated on source host. A CPU scheduling mechanism is used to control the log generation rate. The log files contain the recent running information of the VM. The experimental study presented in [9] derives that with this approach reduction in downtime and total migration time is 62.12% and 43.84% respectively.

Full system trace and replay [10] - Instead of transferring data, system traces and replays events start by taking checkpoint of the source VM and are transferred to destination VM. Simultaneously, source host starts to record non deterministic events (such as input and time variables) into log file which are subsequently sent to destination. The destination replays the log from the mechanism which is faster than the original trace of events. The tracing and replaying goes on until destination machine has a sufficiently small log and the last stop and copy phase is executed. This approach can be challenged in the case of multiple VM migration as total migration time is increased.

Time series based approach [8] – system maintains bitmap of size N. This array records last N history of pages. On every iteration, the page is checked if it is modified or not. The page is declared as high dirty page based upon the threshold value which can be defined by the number of modifications for a particular page. These high dirty pages are sent to the last round in this approach. Performance of this approach depends upon the following parameters: threshold and size of time series array N.

Memory compression [11] - It is an adaptive memory compression technique where memory sent is first compressed at the source and then decompressed at the destination machine. The comparative study in [11] gives us these metrics- reduction in downtime, total migration and total data transferred is 27%, 32% and 68% respectively.

Delta compression [12] – this algorithm is a modification to the KVM hypervisor. The performance is evaluated by migrating running VMs with different types of workload. They demonstrate that when VMs migrate with high workloads and/or over low-bandwidth networks there is a high risk of service interruption. Using delta compression, risk of service can be reduced as data is stored in the form of changes between versions. In order to improve performance, either the dirtying rate has to be reduced or the network throughput increased.

Pre-copy approach based on Least recently used (LRU) policy – migration downtime depends upon amount of data transferred during migration. The working set prediction algorithm approach is based on least recently used (LRU) policy [13]. It results in working sets which collects the most recent used memory pages and inactive list that collects the least recent used memory pages. In this framework, it transfers only inactive list in stop and copy operation. Working set and CPU state is transferred. Total migration time is reduced when compared with pre copy based approach.

Pre-copy approach based on LRU with Splay tree [14] - To reduce the downtime and total migration time, Live VM using LRU with Splay Tree Algorithm technique is introduced. The system defines working set list using splay tree algorithm which collects the most recent used memory pages- pre processing phase. It transfer memory pages expect working set – push phase. The CPU state and working set are transferred after suspending the source VM and the target VM is then activated – stop and copy phase. Result of comparison done with different workloads show that we can reduce 11.45% of total migration time on avg. in Live VM than on XEN.

Dynamic page transfer recording and compression [15] - When migrating CPU/memory intensive VMs, extended migration downtime and prolonged total migration time – these two factors create problems for migration over slower networks. The combination of 1. Reducing the risk of retransfer for frequently dirtied pages. As amount of data is reduced, the total migration time is shortened. 2. The

memory pages are compressed that increase the migration throughput and decrease the migration downtime. Experimental results in [15] states that this combined pre copy approach lessens migration downtime by factor 10 to 20, reduces total migration time by around 35%.

IV. COMPARISON

Different Pre-Copy live VM migration techniques are compared based on different performance metrics, i.e. Downtime, total migration time, total data transferred, bandwidth, workload etc. The following metrics are usually used to measure the performance of live migration [16]:

a. ***Preparation Time:*** The time when migration has started and transfer of VM's states to target host happens.
b. ***Downtime:*** The time when the VM has stopped executing.
c. ***Resume Time*****:** The time between resuming VM on target node and end of migration process.
d. ***Pages Transferred:*** This is total amount of memory transferred, including dirty memory during whole migration process.
e. ***Total Migration Time:*** Sum total of all of the above mentioned times. Time between start of migration and end of migration process.

Table 1. Comparison of different Pre-Copy based live Migration techniques**.**

Sr. No.	***Virtual Machine Migration Techniques***	***Concept***	***Pros***	***Cons***
1	Stop-and-copy approach[17]	Halting the target VM, copying all the pages to the destination and then starting the execution of VM.	1. Simplicity.	1. More service downtime
2	Improved Pre-Copy approach[7]	System records the frequently modified pages in bitmap.	1. Reduces number of iterations and total data transferred. 2. Total migration time is reduced.	1. System downtime is increased.
3	Pre-Copy based on modification of pages [8]	Proposed algorithm divides pages into high dirty pages and low dirty page. High dirty pages are transferred in last phase.	1. Avoid unnecessary transfer of dirty pages. 2. Works in both high dirty pages and low dirty pages situations.	1. Not suitable for wide area live virtual machine migration.
4	Recovery and CPU scheduling [9]	Approach based on recovering system and CPU scheduling.	1. Reduction in downtime and total migration time in comparison with pre-copy algorithm.	1. It is not sure that this approach works well in a more complex environment.
5	Full system trace and replay [10]	Approach based on system tracing and replaying the events occurred during migration.	1. Downtime is reduced by 72.4%.	1. Application performance overhead is 8%.
6	Time series based approach [8]	It identifies high dirty pages from history more precisely and transfers them in the last round of iteration.	1. Number of iterations is decreased. 2. Down time and migration time is reduced. 3. Fewer pages transferred.	1. Useful only when high dirty pages are in the system.
7	Memory Compression [11]	Compression of memory pages to reduce amount of data transferred.	1. Reduction in downtime and total migration time by 27% and 32% respectively. 2. 68% reduction in data transferred.	1. Compression operations introduce additional overhead.
8	Delta Compression [12]	Modification to the KVM hypervisor	1. Downtime is reduced. 2. Increases migration throughput.	1. Compression operations introduced additional overhead.
9	Approach based on Least recently used (LRU) policy [13]	Defines working sets which collect most recently used memory pages and transfers them into last phase.	1. Fewer number of iterations. 2. Reduction in total migration time.	1. For less no of working sets (<1024 pages) the preprocessing phase can be an overhead for total migration time.
10	Approach based on LRU policy with Splay Tree [14]	Same as LRU based approach but the working set is defined with Splay tree algorithm.	1. Reduction in the amount of transferred data by 23.67%. 2. 11.45% reduction of total migration time.	
11	Dynamic page transfer recording and compression [15]	Combination of two technologies, modification of pages and compression.	1. Reduces downtime by a factor 10 to 20, shortens total migration time around 35%. 2. Uses 39% less bandwidth.	1. More resource consumption which harms overall system performance.

V. CONCLUSION

This paper includes a survey of various pre copy based live VM migration techniques. Live migration involves transferring running VM from host to target machine. There are many approaches which attempt to minimize the downtime and total migration time to improve system performance. The comparative study between different pre copy techniques decides which technique we can use in different situations i.e. heavy workload, low bandwidth etc. On comparing different techniques, following results are concluded. Approaches based on bitmap concept - time series based approach and improved pre copy approach are better to reduce number of iterations. Recovery and CPU scheduling reduces total migration time and downtime though it won't work in a complex environment. Full system and trace replay is not good for multiple virtual machine migration. The memory compression technique is better in reducing total migration time, downtime and amount of data transferred. LRU with splay tree technique is better in reducing both total migration time and downtime. For a more complex environment, dynamic page transfer recording and compression techniques are better.

VI. REFERENCES

[1]. "Virtualization" Wikipedia. [online]. Available: http://en.wikipedia.org/wiki/Virtualization [Accessed: March 2014].

[2]. Divya K. Emmanual P. and Ramesh J., "Live virtual machine migration techniques: survey and research challanges" 3rd IEEE International Advance Computing Conference (IACC) 2013.

[3]. Jhon Wiley & Sons, Inc, "How to use Virtualization with cloud computing", [online]. Available: http://www.dummies.com/howto/content/how-to-use-virtulization-with-cloud-computing.html [Accessed: March 2014].

[4]. Spiceworks Inc.,"Understanding the benefits of virtualization ", [online]. Available: community. spiceworks. com/how_to/show/42797-understanding-the-benefits-of-virtualization#conclusion [Accessed: March 2014].

[5]. Tech Target, "Live Migration", [online] http://serachservervirtulization.techtarget.com/defination/live-migration [Accessed: March 2014].

[6]. "Live Migration", Wikipedia. [online]. Available: http://en.wikipedia.org/wiki/Live_migration [Accessed: March 2014].

[7]. Sushil Kumar S. and RaviKant K., "A survey on pre-copy based live migration of virtual machine in cloud environment", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 10, October 2013.

[8]. Sushil Kumar S. and Ravi Kant K., "Enhanced live migration of virtual machine using comparison of modified and unmodified pages ", International Journal of Computer Science and Mobile Computing, IJCSMC, Vol.3, Issue. 2,February 2014.pg.779-787.

[9]. Weining L. and Tao F., "Live migration of virtual machine based on recovering system and CPU scheduling", IEEE 2011.

[10]. Haikun L., Hai J., Xiaofei L., Liting H., Chen Y., "Live migration of virtual machine based on full system trace and reply", ACM 2009.

[11]. Erik G. "Optimizing total migration time in virtual machine live migration" Department of Information Technology, Uppsala University.

[12]. Petter S., Benoit H., John T., Erik E., "Evaluation of delta compression techniques for efficient live migration of large virtual machines", 7th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, California, USA:ACM,2011.

[13]. Ei Phyu Z., and Ni Lar T.,"Live virtual machine migration with efficient working set prediction" International Conference on Network and Electronics Engineering, Singapore, 2011.

[14]. Ei Phyu Z., and Ni Lar T.,"Improved live VM migration using LRU and splay tree algorithm" International Journal of Computer Science and Telecommunications, Volume 3,Issue 3,March 2012.

[15]. Petter S., John T., Benoit H., Erik E., "High performance live migration through dynamic page transfer recording and compression" 3rd IEEE International Conference on Cloud Computing Technology and Science, 2011.

[16]. Michel H., Umesh D., Kartik G., "Post-copy live migration of virtual machines", SIGOPS Oper. Syst. Rev., Vol.43,pp.14-26,2009.

[17]. Gulshan S. and Mala K., "Comparative study of live virtual machine migration techniques in cloud" International Journal of Computer Applications (0975-8887) Volume 84- No 14, December 2013.