Survey on Different Virtual Machine Migration Approaches

Praveen Jain[1], Ratish Agarwal[2], Sachin Goyal[3]
[1] S.O.I.T, R.G.P.V. Bhopal, India
[2, 3] U.I.T., R.G.P.V. Bhopal, India

Abstract- *"Cloud Computing" is considered to be one of the most important innovation of the modern IT industry. Cloud computing is a rapidly evolving model in which new aspects and capabilities are introducing regularly "Virtualization" is a foundational technology platform fostering cloud computing. Virtualization refers to the abstraction of the computer resources from applications and end users consuming the service. It minimizes the initial and maintenance cost of the data centres by implementing different Virtual Machines (VM) on the same Physical Machines (PM). VM migration can be performed to minimize energy consumption, load balancing, fault management, low-level system maintenance. VM migration are use to migrate OS instances (Processor State, Storage, Memory, Network Connection) from one PM to another PM on a cloud environment. Different techniques are available for the migration of the VM data from one PM to another named Pre-copy, Post-copy and CR/TR-Motion, Live migration using CPU Scheduling, Resource reservation based multiple VMs Live migration. These all methods have different down time and total migration time which directly affect the system performance. Hence, prime objective of live migration approaches is to minimize the downtime and total migration times. This paper gives the comparative study of varies existing load balancing approach with their anomalies.*

I. INTRODUCTION

"Cloud Computing" is the latest evolution of computing which provides capability to use computing and storage resources and reduce The investments in an organization's computing infrastructure. Cloud Computing is Based on model [2] in which resources are shared unlike previous computing models in which resources are dedicated to a single user. It Provide the scalability, elasticity, pay as u go, resource pooling, on-demand self service to its users, there are different models[4,5] in cloud like.

Cloud Service Models

Software as a Service (SaaS): National Institute of Standards and Technology (NIST) define Cloud Software as a Service (SaaS) as follows: "The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a client interface such as a web browser (e.g. web-based e-mail). The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings." Examples are Zoho Suite, Apple's Mobile Me, Google Docs.

Platform as a Service (PaaS): In Platform as a Service model vendor provides development environment for the application developers and can be access through web browser, Paas also offers the developers to build web applications without require specialized expertise. Vendor examples are Google App Engine, force.com (from salesforce.com), Microsoft Azure.

Infrastructure as a Service (IaaS): In Iaas model,vendor provides the on demand infrastructure to the customer to run the applications, the main features of Iaas model are pay as you go, scalability. Amazon offers a large menu of cloud services for example

- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Simple DB
- Amazon Simple Storage Service (Amazon S3)
- Amazon Simple Queue Service (Amazon SQS)

Virtualization

Virtualization is the key concept of the cloud computing, virtualization is the abstraction of the physical resources (hardware) where multiple operating systems (guests) can run concurrently on the same physical host machine, it has improved the serviceability and availability of the resources.

Virtualization provides different key benefits such as Utility pricing, Hardware independence, Scalability, Usage based on SLA, Fault Tolerance

Virtual Machine Monitor (VMM)

Virtual machine Monitor (VMM) or hypervisor is a program that is used to attain virtualization in a cloud environment, there can be many VM on a single PM, all VM

share underlying physical resources of a PM, virtualization is gaining popularity as it provide the advantages like

- Server consolidation
- Resource isolation
- Live migration

There are different kinds of virtualizations like Server virtualization, storage virtualization, OS virtualization.

Virtual Machine Migration

virtual machine migration is the process to migrate VM from one PM to another under different conditions like for consuming power or in balancing the load of PM, there are different kinds of migration techniques like pre copy, post copy, CR/TR-Motion, Live migration using CPU Scheduling, Resource reservation based multiple VMs Live migration there can be two types of VM migration one is live migration and other is non live, in live migration VM does not loses its state, and there will be no interruption in the services provided by it but in cold or non live migration VM loses its state and interruption can be seen.

II. LIVE VM MIGRATION TECHNIQUES

2.1 Pre copy: In Precopy [7] approach first memory pages are copied from source to destination without suspending the VM and then the dirty pages created during copying will be copied, the copying of dirty pages depends on the number of iteration used in the algorithm , two metrics are used to measure the performance of live migration one is 'downtime' this is the time taken in which the VM is unavailable for the users and other is the 'total migration time' this is the time taken to migrate the system state and memory from source to destination. Down time is proportional to the size of VM to be transfer.

2.2 Post copy: In this technique first cpu state is transferred from source to destination after suspending the source node then the pages can be fetch from source to destination also called the demand paging , in this technique downtime is minimal but the total migration time is infinite. There are different variants of post copy method [17] depends on fetching of page like dynamic self-ballooning (DSB), active push, pre paging and demand paging.

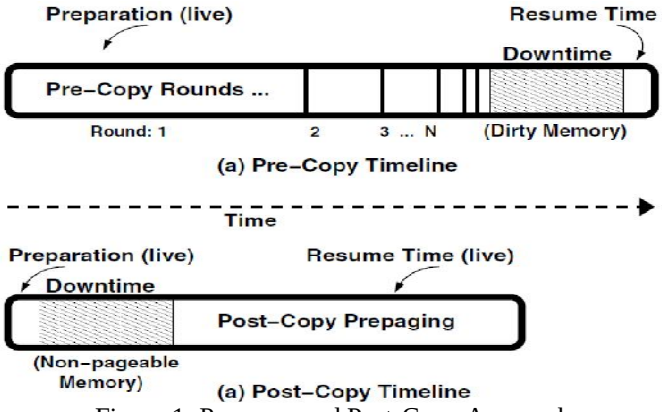

Figure 1: Pre-copy and Post-Copy Approach

2.3 CR/TR-Motion: Check Point Recovery/Trace and Replay technology [12] provides very fast and transparent VM migration because it reduces the network traffic by generating small size logs on the source node, in this technique log file generated on the source host is transferred to the target host iteratively. Check point/recovery and trace/replay are widely used for system state recovery due to failure, dirty memory pages are not copied in this method this also decrease the amount of data transferred only the log files are synchronised at both source and destination, log transfer rate is greater than the log generate rate which makes this method effective, this method also having a problem that the source and destination VMs should have same OS running because logs are different for different for different OS

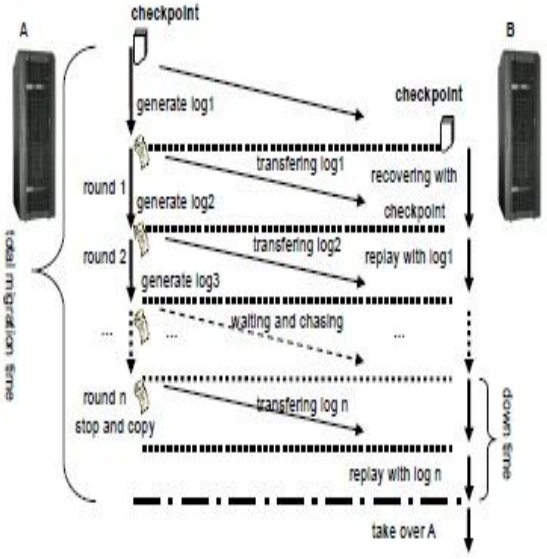

Figure 2: Process of live migration using CR/TR-MOTION

2.4 Live migration using CPU Scheduling:

In this method the number of pages to be transferred are reduced by cutting down the CPU performance, hence the CPU speed is reduced the dirty pages are also less, the main

advantage of this approach are the less dirtied pages to be transferred but also comes with the disadvantage of degraded CPU performance

2.5 Resource reservation based multiple VM Live migration:

This approach is used where there is a need to migrate more than one VM i.e. multiple VMs are migrated to different target using resource reservation [14], multiple VM migration faces many problems like migration failure due to insufficient resource available at the destination host, migration thrashing and migration conflicts. But resource reservation methods attempt to solve many of the above problems. Resource reservation method consists of four main modules: Migration Decision-Maker, Migration Controller, Resource Reservation Controller, and Resource Monitor. It should be implemented both on source (CPU and memory resource) and at destination includes the whole VM resources.

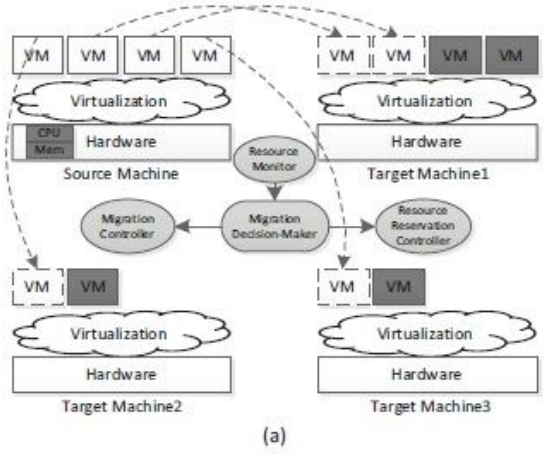

Figure 3 (a): Live migration framework of multiple VM using resource reservation technology

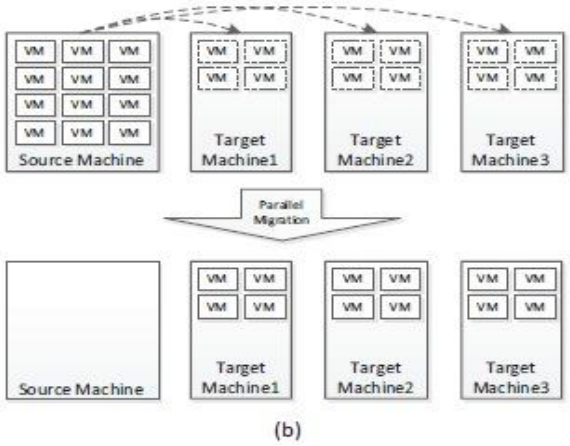

Figure 3 (b): Shows parallel live migration of multiple VMs

MIGRATION GOALS

Load balancing: It can be accomplished by migrating VM's from overloaded PM to under loaded PM, This can be done to reduce inequality among several PMs in a cluster.

Server Consolidation: Server consolidation algorithms are used to reduce server sprawl, it improves the resource usage by turning off the unused or underutilized machines thereby reducing the energy consumption and overall operational cost of the data centres.

Hotspot & Coldspot Mitigation:

The hotspot and cold spot are the threshold values which are set by data centre owner or as described in the SLA specified by the clients upper and lower threshold are higher resource usage and lower resource usage, hotspot are formed when the usage of PM is beyond (over utilization) the upper threshold and coldspot are formed when the usage are below (under utilization) the lower threshold in above all the conditions migrations are triggered.

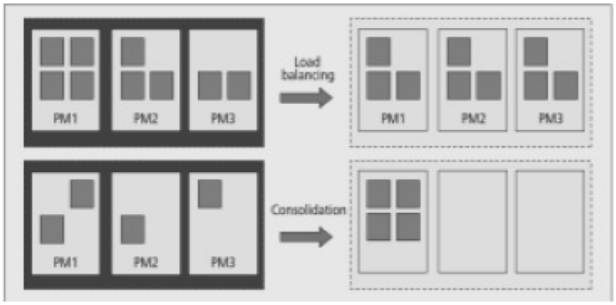

Figure 4: Load balancing and server consolidation

Heuristics for resource management using migration

To accomplish all the three goals (server consolidation, load balancing and hotspot Mitigation) of VM migration there is a need to address some questions like

When to migrate the VMs

There are many different situations when there is a need to migrate VMs such as periodic, load imbalance, due to hotspot, excess spare capacity,

Which VMs to migrate

Selection of candidate VM is one of the crucial decision of the resource management because it not only makes the VMs unavailable but also consumes the network and CPU both at source and destination node there are

different methods to select which VMs to migrate like resource constrained virtual machine, holistic approach, affinity based approach.

Where to migrate the VMs

Selection of destination is also important in the live VM migration process depends on whether the destination PM has enough resources to handle the incoming VM or depends on affinity of VMs.

III. NOTEWORTHY CONTRIBUTIONS

R. Michael et al. [17] proposed live migration approach based on the post copy. Post-copy migration defers the transfer of a VM's memory contents until after its processor state has been sent to the target host. This deferral is in contrast to the traditional pre-copy approach, which first copies the memory state over multiple iterations followed by a final transfer of the processor state. They use Dynamic Self-Ballooning (DSB) to reduced the number of free pages without significantly impacting the normal execution of the VM, so that the VM can be migrated quickly with a minimal memory footprint.

Problem: - Once the VM resumes at the target, demand paging begins for missing pages that can increase the total migration time.

F. Ma et al. [7], proposed Improved Pre-copy approach for migrating the VM in which some bitmaps are introduced which marks frequently updated pages . Although pre-copy approach can balance the downtime and total migration time of contradictions, it will cap the number of copying iterations to the maximum number of iterations. In this paper, we improve the pre-copy approach on Xen 3.3.0.. By the judge in the iteration process, we put those frequently updated pages into the page bitmap, and those pages can only be transmitted in the last round of the iteration process. This ensures that those frequently updated pages are transmitted just once in the iteration process. According to the used state of the page, the memory page in Xen which will be migrated is divided into three categories, i.e. three kinds of bitmap page of Xen: TO_SEND, TO_SKIP and TO_FIX. The three kinds of bitmap page are described as following:

TO_SEND: - Marked the pages which get dirty in the previous iteration process, i.e. the pages which need to be transmitted in this iteration.

TO_SKIP: - Marked the pages which can be skipped in this iteration.

TO_FIX: - Marked the pages which need to be transmitted in the last iteration.

In fact, the pages in the TO_SEND bitmap page are the pages which are modified in the previous iteration process. Those frequently updated pages recorded in TO_SKIP bitmap page, and those pages which have not been mapped, in TO_FIX bitmap page, and be transmitted in the last round of the iteration process. And, we add a bitmap page named TO_SEND_LAST in our improved pre-copy approach. As the TO_FIX bitmap page, TO_SEND_LAST marks the pages which need to be transmitted in the last iteration. However, unlike TO_FIX, TO_SEND_LAST records those frequently updated pages. How to determine those frequently updated pages? In order to facilitate control, we think if it appears a retransmission of the page, the page should be put into the TO_SEND_LAST bitmap page, i.e. the page which appears in the TO_SKIP bitmap page should also be put into the TO_SEND_LAST bitmap page.

Problem: - This approach minimized the down time, but can be more optimized because page is declared as a frequently updated if it is modified in last two iterations.

Wentian cui and M`eina song [16], proposed an VM migration approach by using Matrix Bitmap Algorithm. This approach is based on the "Program Locality Principle", and use "matrix bitmap algorithm" that collects the dirty page information for many times before deciding whether to transfer the page or not. It provides a more reasonable approach to obtain the determination. The primary imperfection of Pre-copy is that the information gathered to predict whether a certain page. will be modified later is beyond adequacy. Since the two dirty bitmaps, to_send and to_skip, are insufficient, why not collect the dirty bit map for many times and then decide whether to send or not for one time. It's the core ideology of "matrix bitmap algorithm".
Algorithm for the method

1. MAP_LEN, a constant quantity. Set by the system administrator to define the number of dirty bitmaps collected at one iteration.

2. dirty_bit_map[MAP_LEN], matrix bitmap with the width of MAP_LEN. It is applied to collect the dirty bitmap for MAP_LEN times and then decide whether to send or not for one iteration. The files of this bitmap demonstrate the modifying conditions of the corresponding pages.

3. dirty_bit_map_tag, page bit map tag. It is an essential variable initialized to 0 that manipulates the page transference, which is used to record the situations of the pages (modified,

sent). When a certain page is modified, its corresponding dirty_bit_map_tag is set as 1, which means this page may be sent at this iteration. In the case this page is transferred, the corresponding dirty_bit_map_tag will be cleared.

4. time_slot, to get the dirty bitmap every time_slot milliseconds. The time_slot is initialized to 900, for the subsequent iteration, it decreases 80 every time. The values 900 and 80 are calculated from xen3.4.2 using the actual data in its xend.log.

5. weight, page send weight. It is calculated from dirty_bit_map[MAP_LEN]. For instance, MAP_LEN=4 and a certain page is modified like this: 1011. To read from left to right, then weight= (1101)2= (13)10.

6. THRESHOLD, page send threshold. If weight>= THRESHOLD, this page won't be sent in this iteration. THRESHOLD is set as [2^(MAP_LEN-1)] here. The below Table shows the Page Transfer Strategy when MAP_LEN=4 and THRESHOLD=8. The red columns demonstrate the superiority to Pre-copy. In addition, the value of THRESHOLD is flexible, but distinctly, the smaller value indicates the increase of downtime, while the larger one will result in the excessive total migration time.

Table: Page Transfer Strategy

Dirty	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Bitmap	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
Collected	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
4 times	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Send or not	N	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N

Problem: - If the page is 1010, then it will be transferred, which will increase the total migration time.

H. Bolin et al. [9], proposed a pre copy based live migration technique for the cloud. In this paper, they evaluate traditional precopy approach, and propose an improved time-series based precopy approach for virtual machine migration. With the time-series prediction technique, they identify frequently updated dirty pages (high dirty pages) in the past and future period more precisely, and transmit them in the last round of iteration, in order to reduce unnecessary, repeated transmission of dirty pages. Doing so can significantly reduce the total migration time.

They propose a time-series based precopy algorithm. In this algorithm, we identify high dirty pages in iteration by their historical statistics of the TO_SEND bitmap, to avoid them being transferred again to the target host in this iteration. Until at the end of the last iteration, will be these high dirty pages transferred to target host. The benefits brought in are obvious: the number of iterations of migration decreases, the down time and migration time also decrease. In other words, the VM can be moved to the target host much faster.They introduce two key parameters K and N, and an array of historical bitmap called TO_SEND_H with size N. The TO_SEND_H is used to save the past statistics of the TO_SEND bitmap. The N stands for the max size of the time-series array TO_SEND_H, and the K stands for the threshold of high dirty page, which means only dirty pages whose appearing times in the TO_SEND_H array passing K can be identified as high dirty pages. During the process of iteration, the data of the TO_SEND bitmap will be saved to array TO_SEND_H. Following equation is use to identifying high dirty page p:

$$\sum_{i=1}^{N} (p \in to\_send\_h[i] \geq K)$$

Problem: - Identification of the value of k and n is very difficult.

Ibrahim et al. [18] presented an algorithm that achieves both low downtime and low application performance impact. This algorithm detects memory update pattern and terminate migration when there is an occurrence of improvements in the downtime. They implemented this approach in KVM and demonstrated its benefits for both Ethernet and RDMA (InfiniBand) migration.

Soramichi et al. [19] discuss the placement of Virtual Machines (VMs) on physical hosts in which image of the virtual machine which is to be migrated are kept on this host only, when there is a need of migration from another host to that host (which are having image) the kept memory image will be reused. The kept pages will not be transferred which saves the amount of data to be transferred , this also prevents dynamic consolidation systems from optimizing VM placements efficiently.

The proposed technique called "memory reusing " reduces the amount of transferred memory of live migration. Implemented a system named Miyako Dori that uses memory reusing in live migrations.

IV. CONCLUSION

After reviewing the theory of VM migration it is found that method which is use to transfer the VM data from one PM to another play a vital role in system performance. System performance is mainly depends on the downtime and total migration time. This paper presents the use of virtualization in cloud computing technology and different live

VM migration techniques. Main objective of all migration techniques are to reduce the downtime and total migration time and to provide the better performance in lower bandwidth. Different techniques have strengths and weakness with some extent, modern data centers are now moving to virtualized environment so there is a greater need of VM migration to improve the efficiency of the data centers.

REFRENCES

[1] R. Hunter, The why of cloud, http://www.gartner.com/ DisplayDocument?doc cd=226469&ref= g noreg, 2012.

[2] M. D. Dikaiakos, G. Pallis, D. Katsa, P. Mehra, and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", IEEE Journal of Internet Computing, Vol. 13, No. 5, pp 10-13, 2009.

[3] P. Mell and T. Grance. The NIST definition of cloud computing (draft).NIST special Publication,, pp.273–86, 2011.

[4] R. K. Gupta et al., "A Complete Theoretical Review on Virtual Machine Migration in Cloud Environment", International Journal of Cloud Computing and Services Science (IJ-CLOSER), Vol.3, No.3, pp. 172-178, 2014.

[5] B. P. Rima et al., "A Taxonomy and Survey of Cloud Computing Systems", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, pp. 44-51, 2009.

[6] P. Kaur and A. Rani , "Virtual Machine Migration in Cloud Computing", International Journal of Grid Distribution Computing, Vol. 8, No.5, pp.337-342, 2015.

[7] F. Ma, F. Liu, Z. Liu, "Live Virtual Machine Migration based on Improved Pre-copy Approach", IEEE, Vol.7 No.10, pp-230-233,2010.

[8] S. Kaur, V. Pandey , "A Survey of Virtual Machine Migration Techniques in Cloud", Computer Engineering and Intelligent Systems, Vol.6, No.7, pp-28-34,2015.

[9] B. Hu, Z. Lei, Y. Lei, D. Xu, J. Lei, "A Time-Series Based Precopy Approach for Live Migration of Virtual Machines", IEEE 17th International Conference on Parallel and Distributed Systems, pp-947-952, 2011.

[10] J. H. Sheth, K. N. Vaghela, "Technical Review on Live Virtual Machine Migration Techniques for Eucalyptus Cloud", Int. Journal of Engineering Research and Applications, Vol. 5, Issue 3 (part-2), pp.50-52,2015.

[11] D. Kapil, E. S. Pilli and R. C. Joshi," Live Virtual Machine Migration Techniques:Survey and Research Challenges", 3rd IEEE International Advance Computing Conference (IACC), pp-963-969,2013.

[12] H. Liu, H. Jin, X. Liao, L. Hu, C. Yu, "Live Migration of Virtual Machine Based on Full System Trace and Replay",HPDC'09, June11–13, 2009, Munich, Germany.Copyright2009 ACM 978-1-60558-587-1/09/06.

[13] R. K. Raj, G.Jeba, Leelipushpam.P, "live virtual machine migration Techniques–a survey", International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 7, September – 2012.

[14] K. Ye, X. Jiang, D. Huang, J. Chen, B. Wang," Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments", IEEE 4th International Conference on Cloud Computing,2011,PP 267-274.

[15] A. Mohan, Shine S, "Survey on Live VM Migration Techniques", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 2, Issue 1, January 2013, pp155-157.

[16] W. Cui, M. Song, "Live Memory Migration with Matrix Bitmap Algorithm", IEEE, 978-1-4244-6359-6/10,pp 277-281,2010.

[17] M. R. Hines and K. Gopalan," Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self-Ballooning", ACM, 978-1-60558-375-4/09/03,PP 51-60, March 11–13, 2009.

[18] K. Z. Ibrahim, S. Hofmeyr, C. Iancu, and E. Roman 2011. Optimized precopy live migration for memoryintensive applications. International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 1-11.

[19] Soramichi Akiyama, Takahiro Hirofuchi, Ryousei Takano and Shinichi Honiden,"MiyakoDori: A Memory

Reusing Mechanism for Dynamic VM Consolidation", 2012 IEEE Fifth International Conference on CloudComputing