

A Survey on the Live Migration of Virtual Machines

Nikhil Karkare

Graduate Student

University of Texas at Tyler

ABSTRACT

Live migration is a technology used to move a virtual machine running on one physical machine to another with unnoticeable disruption to the clients. This technology helps the cloud service providers to provide 'pay as you go' service. Main issues to overcome are the downtime, total migration time, efficiency in LANs and WANs and energy consumption during migration. In this paper, we have surveyed different approaches used for live migration, categorized them and compared them to check their efficiency in all the categories listed.

1.INTRODUCTION

Live migration means moving the virtual machines from one physical machine to another without disconnecting the clients and applications. It is mainly used by the cloud service providers [5]. Nowadays, the cost for the maintenance of servers is increasing gradually. This is the reason that the companies are reducing their number of servers and switching to the cloud services which only costs them as much as they use [4]. So, it is the biggest challenge for cloud service providers to provide services to their clients efficiently and without any disruption. Sometimes the data has to be migrated because if the data center becomes unavailable due to maintenance, security failures or any catastrophic events, then clients will be disconnected. But, client must always be connected. Live migration is used as a powerful tool to achieve this objective.

Live migration of virtual machines (VM) is done by two methods traditionally, pre-copy and post-copy memory migration. These techniques are good to reduce the downtime but the total migration time increases at the same time [3]. So, efficiency of these prevalent techniques should be improved. In this paper, the approaches used for live migration are categorized into four categories. These categories mainly focus on the downtime and the total

migration time. When the migration is done in LAN's (Local Area Networks), run-time memory state of VM is transferred. And in case of WAN's (Wide Area Networks), its file system and whole network connections are transferred [6]. A capability for migrating live VMs among multiple distributed sites provides a significant new benefit for VM [7]. VM is also vulnerable to some attacks. So, the source and destination platforms are trusted, migration data should remain confidential and unmodified during transmission [9]. There are four main factors that should be kept in mind while migrating the virtual machine from source to destination physical machine: downtime (i.e. time during VM is in idle state) and total migration time (i.e. the time from which migration starts until it starts running on the target host and the data on the source host is destroyed), network bandwidth usage, cost and security issues.

The main problems caused by the traditional techniques are more downtime, more migration time, more network bandwidth consumption and more energy consumption. All these factors should be reduced to achieve efficient live migration. The energy cost of virtual machine live migration consists of the energy additionally consumed by the source as well as the target host [8]. These problems are negligible when the amount of data to be migrated is less. But, when it comes to gigabytes, above problems can easily be noticed. It can directly affect the connectivity between the clients and server. It is a big challenge for cloud service provider to achieve negligible downtime so that the clients doesn't get disconnected. There are lot of clients connected to the server at the same time. If something goes wrong then the clients cannot access anything on that server until the migration is done. The objective of this survey paper is to discuss the core idea of different approaches used for live migration of virtual machines and compare them.

In this paper, four approaches are studied and compared. Approaches in the first category mainly concerned about reducing downtime only. Only approach in this category is three-phase migration (TPM) and Incremental Migration (IM). These are the algorithms focuses on reducing the downtime. Second category includes the approaches which reduces the total migration time only. In this category, an approach studied reduces the migration time by adding a storage device (NAS) between source and the target host. In some cases of this approach downtime gets increases. Another approach in this category improves the efficiency of pre-copy approach by using a bitmap. Also it contains an approach which uses LRU (Least Recently Used) and splay tree algorithm in the live migration to reduce the number of pages to be transferred. Third category approaches are more efficient than the approaches in previous two categories as they reduces both the downtime and total migration time. In this category, one approach is based on a technology called check-pointing/recovery and trace/replay (CR/TR-Motion) which is used for live migration. This approach is used for live migration in LAN environments and reduces the network bandwidth consumption. There is another approach in this category which uses block-level solution and it is most efficient. This category also has some techniques that eliminates the duplication of pages, and in the last approach data is compressed and decompressed at the source and destination respectively. Last category focuses on some additional factors like energy consumption. The approach studied in this category is the implementation of live migration feature to the Eucalyptus, an open-source cloud computing environment. It reduces energy consumption. All these approaches have some advantages and disadvantages when compared and are explained in the next section.

2. APPROACHES

In this part, all the approaches are categorized into four categories. These categories are explained first and then compared.

2.1 Approaches that Reduce Downtime Only

2.1.1 Three-phase migration (TPM) and Incremental Migration (IM):

It is a good and efficient solution which can reduce the downtime during the migration. Generally there are two methods to migrate the virtual machine, pre-copy and post-copy. Solution combines these two methods in one algorithm and add one more phase to it, which is called freeze and copy phase. This algorithm is called Three Phase Migration (TPM).

First phase is called pre-copy and in this phase, all the storage data is pre-copied iteratively to the destination. If the rate of dirty pages generation gets higher than the transfer rate then this phase stops and there is always a limit for the iteration. Now comes the most important phase which is called freeze and copy. Here, the most important entity of the paper which is "block bitmap" is used. In this phase, only the important data is migrated. The remaining data is fetched when a request comes. Block bitmap keeps the note of all the dirty data which is then sent to the destination. In the next phase which is called post-copy phase, the virtual machine starts running and fetches the dirty data through bitmap. As the bitmap transfers only dirty data, so it decreases the downtime. Block bitmap itself is small in size. So, there is a negligible time consumed by the bitmap transfer.

Another algorithm introduced is called incremental algorithm (IM) which is used when the data is to be migrated back to the source. A difference is always maintained between source and destination. So, this difference only gets migrated back which consumes very less time and the virtual machine starts running on the source once again. This mechanism reduces the migration overhead and IM reduces the synchronization time [1].

2.2 Approaches that Reduce Total Migration Time Only

2.2.1 Virtual Machine Migration Using Shared Storage:

In this technique, Network-attached storage (NAS) device is used as a shared storage device which maintains an updated mapping of memory pages that currently reside in identical form on the storage device. The host which runs the virtual machine has permanent memory and the cache memory. The operating system and running applications occupy some part of the memory and rest of the space remains unused. Modern operating systems use this

unused memory to cache recently accessed blocks of the attached storage device. The data of this cache is thus duplicated: one copy resides on the permanent storage device, another copy exists in the memory of the VM. So, in network-attached storage device an updated list of memory pages is maintained. When VM migrates, this data is fetched directly from NAS rather than fetching from the source. This results in the reduction of the total migration time [3].

2.2.2 Improved Pre-copy Approach

This approach is an advanced version of the traditional pre-copy approach. In pre-copy approach, the data gets migrated iteratively [11]. These iterations are more in number (30 approximately). To lower down the number of iterations, a bitmap is used. This bitmap keeps the note of frequently updated data and migrate this data to the destination in the last round of iteration. Improved pre-copy approach reduces the number of iterations (maximum 5 iterations are done to migrate a VM). So, the total migration time get reduced but the downtime increases because duplicate pages were placed in the last round of the transmission. As the migration time decreases, energy consumption will also reduce.

2.2.3 LRU and Splay Tree Algorithm

In this algorithm, stacks and counters are used. Top of the stack contains last recently used pages. This algorithm consists of three steps: 1) pre-processing, 2) push phase and 3) stop and copy phase[12]. During pre-processing phase it calculates the recently used memory pages. The pages that are not recently used, are transferred to the push phase. Now the dirtied pages get transferred iteratively to the destination. In stop and copy phase, virtual machine stops running on the source and resumes at the destination host. As the less number of pages are transferred during migration, total migration time is reduced.

2.3 Approaches that Reduce Both Downtime and Total Migration Time

2.3.1 Check-pointing/recovery and trace/replay (CR/TR-Motion):

In this algorithm, a check-pointing buffer and logs are used. This migration starts from the selection of the proper target host which can guarantee migration. When this target host is selected, the

source host freezes and copies all its system state information to a check-pointing buffer. Migration starts from here when this buffer transfer the data to the target host. The moment at which target host receive the information, virtual machine starts running on it. All the further events are transferred via the log files. These logs generates and transfers. But, the rate of transfer is always greater than the rate of generation of log files. As the log files transfers, its size keeps decreasing in every next transfer and its gets replayed on the target host. When all the files gets transferred, virtual machine on the source host gets suspended. But, still the source host will be considered as primary host. This algorithm reduces the downtime to 72.4 percent in a LAN and it reduces network traffic in case of WAN [2].

2.3.2 Migration by combining a block-level solution and pre-copying:

This approach consists of four stages. First stage is initialization in which the migration client on the source host connects to the daemon running on the destination host. Daemon accepts the connection request. Now comes the bulk transfer stage. In this stage, VM disk image is transferred to the destination daemon and source continues to run. After bulk transfer the system invokes Xen's live migration interfaces. Xen is a native hypervisor providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently. Xen iteratively logs dirtied memory pages and copies them to the destination host without stopping the VM being migrated. In its final phase it pauses the source VM, copies any remaining pages to the destination, and resumes execution there. During both stages the write operation on the source file system stops and delta generation starts. Delta is a communication unit consist of the written data and they are queued at the destination. These deltas are queued on the destination host. When Xen migration is about to complete, VM is paused and its copy at the destination is started. Because of this the downtime is reduced.

2.3.3 Post-copy Based Migration Using Adaptive Pre-Paging and Dynamic Self-Ballooning

In the traditional post-copy approach, total migration time is very high. If adaptive pre-paging is combined with the post-copy approach, then the duplicate page

transmission can be eliminated [10]. Also the transfer of free memory pages will be eliminated if a dynamic self-ballooning mechanism is added. It is very efficient in LAN's (Local Area Networks) but it doesn't work well in WAN's (Wide Area Networks). It reduces the total migration time and downtime. Downtime is reduced because of shadow paging but it should be lesser.

2.3.4 Virtual Machine Migration Using Adaptive memory Compression

When the network overhead is low, then it is difficult to provide fast migration of virtual machine. To prevent this condition adaptive memory compression is done. Before the transmission of data, it is compressed and transferred to the destination host. At the destination host, data is decompressed. Before compression characteristics of the data is analyzed. They are characterized on the basis of strong and weak regularity[13]. This compression algorithm makes the pages move faster which results in the reduction of total migration time and the downtime.

2.4 Approaches Reducing Energy Consumption

2.4.1 Live Migration in Eucalyptus:

Eucalyptus does not support virtual machine live migration. In this paper, this feature is added to the Eucalyptus. Synchronization between source and destination is done by the distributed replication block device (DRBD), which transfers the disk images between the servers. In this approach, the virtual machines are divided into layers which reduces the amount of data to be transferred. The combination of DRBD and multi-layered root file system is used to reduce energy consumption. Authors have used Advance Configuration and Power Interface (ACPI) which is a prevailing power interface independent of hardware vendor specification. In short, the instance is relocated from source to destination which is initiated by the cluster controller relocation agent and supervised by the node controller on the corresponding server [4].

3. CONTRAST AND COMPARISON

Parameters that are used for the comparison are total migration time, downtime, efficiency in LAN/WAN and energy consumption. One of the parameters

taken for comparison is total migration time. It is the duration from when the migration starts to when the states on both machines are fully synchronized. Three-phase migration (TPM) and Incremental Migration (IM) algorithms doesn't focus more on total migration time. It deals with reducing the downtime between migration. Total migration time is taken into account as a future work of these algorithms. Check-pointing/recovery and trace/replay (CR/TR-Motion) technique reduces the total migration time, when used for the migration in Local Area Networks (LAN's) but in case of Wide Area Networks (WAN's), it doesn't affect the total migration time. VM migration using shared storage mainly focuses on the reduction of total migration time. It uses NAS device, because of which the target host fetches the pages from NAS and the total migration time reduces. Improved pre-copy approach reduces the total migration time because number of iteration reduces. Post-copy migration using Adaptive Pre-paging and Dynamic Self-ballooning reduces the total migration time and it uses the traditional post-copy approach. There is one approach which is taken as an application of live migration as it is implemented in Eucalyptus which results in reducing the energy consumption in the cloud computing environment. Another approach to achieve the objective is to combine the existing pre-copy method and the block-level solution. This approach is most efficient as compared to other approaches. Unlike [2], it reduces the total migration time both in LAN and WAN environments. VM migration by this approach consumes only 3 seconds and 68 seconds in LAN and WAN environments respectively. LRU and splay tree algorithm reduces the total migration time because less number of pages are transferred during migration. VM migration using Adaptive Memory Compression reduces the total migration time because compressed data move faster over the network.

Another point that can be taken for the comparison is downtime. It is the time interval during which services are entirely unavailable to the clients. First approach i.e. three-phase migration (TPM) and Incremental Migration (IM) algorithms mainly focuses on reducing downtime. These algorithms successfully reduces downtime up to 72.4 percent. Check-pointing/recovery and trace/replay (CR/TR-Motion) technique used for live migration reduces the downtime in LAN environment but it reduces

Approaches	Author's Name	Year of Publication	Efficient in LAN's	Efficient in WAN's	Downtime	Total Migration Time	Energy Consumption
TPM and IM ^a	Yingwei Luo et al.	2008	Yes	No	Reduces	Doesn't Focus	More
CR/TR Motion ^b	Haikul Liu et al.	2008	Yes	No	Reduces	Reduces	Less
VM Using Shared Storage	Changyeon Jo et al.	2013	Yes	No	Doesn't Focus	Reduces	Less
Migration in Eucalyptus	Pablo Graubner et al.	2011	Yes	No	Doesn't Focus	Doesn't Focus	Less
Migration Using BS-PC ^c	Robert Bradford et al.	2007	Yes	Yes	Reduces	Reduces	Less
Improved Pre-Copy Migration	Fei Ma et al.	2010	Yes	No	Doesn't Focus	Reduces	More
Post-Copy Migration using APP & DSB ^d	Micheal Hines et al.	2009	Yes	No	Reduces	Reduces	Less
VM Migration with Adaptive Memory Compression	Hai Jin et al.	2009	Yes	No	Reduces	Reduces	Less

^a Three-phase Migration and Incremental Migration

^b Check-pointing/recovery and trace/replay

^c Block-level solution and pre-copying

^d Adaptive Pre-paging and Dynamic Self-ballooning

Table 1: Table showing the comparison between the surveyed approaches

network bandwidth consumption in both LAN and WAN environments. In VM migration using shared storage, it balances the downtime when duplication rate is low. But, when there is more duplication, downtime cannot be controlled. In fourth approach, downtime is not taken into consideration. Approach used in paper [6] reduces the downtime. During the process of migration within a LAN, the VM doesn't stop at all (i.e. the downtime is unnoticeable) and in case of WAN, VM doesn't stop till three phases. It just pauses because whole network has to be migrated to the destination. It also uses the mechanism called write throttling which slows down the write accesses by VM. This helps in reducing the network bandwidth consumption. When it gets compared with freeze and copy phase in three-phase migration (TPM), it proves to be a better approach as in the wide-area, and this approach reduces service disruption by several orders of magnitude. All the approaches works very well in LANs but there is only one approach which works efficiently in WANs, migration using block-level solution and pre-copying. All the approaches that doesn't focus on the total migration time, increases energy consumption.

4. CONCLUSION

Live migration is to move the virtual machine from one physical machine to another. Live virtual machine migration is helpful for the cloud service providers as it saves the server energy consumption and the time to allocate the memory space requested by the clients. There are several migration techniques. Most of the techniques doesn't work efficiently in WANs. Lowest downtime achieved in all the approaches was 3 seconds. . In future, an approach should be introduced, so that VM migration will work very well in WANs and take the downtime in microseconds. By this, the disruption time will be unnoticeable to the clients connected to the virtual machine.

5. REFERENCES

- [1] Yingwei Luo et al. "Live and Incremental Whole-System Migration of Virtual Machines using Block-Bitmap", *2008 IEEE International Conference on Cluster Computing*, pp: 99-106.
- [2] Haikun Liu et al. "Live Virtual Machine Migration via Asynchronous Replication and State Synchronization", *December 2008 IEEE Transactions on Parallel and Distributed Systems*, pp: 1986-1999.
- [3] Changyeon Jo et al. "Efficient Live Migration of Virtual Machines Using Shared Storage", *March 2013 Proceedings of the 9th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*.
- [4] Pablo Graubner et al. "Energy-efficient Management of Virtual Machines in Eucalyptus", *2011 IEEE 4th International Conference on Cloud Computing*, pp: 243-250.
- [5] Ching-Chi Lin et al. "Energy-efficient Virtual Machine Provision Algorithms for Cloud Systems", *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, pp: 81-88.
- [6] Robert Bradford et al. "Live Wide-Area Migration of Virtual Machines Including Local Persistent State", *Proceedings of the 3rd international conference on Virtual execution environments*.
- [7] Franco Travostino et al. "Seamless live migration of virtual machines over the MAN/WAN", *Future Generation Computer Systems, Volume 22, Issue 8, October 2006, Pages 901-907*.
- [8] Anja Strunk "Costs of Virtual Machine Live Migration: A Survey", *2012 IEEE Eighth World Congress on Services*, pp: 323-329.
- [9] Jyoti Shetty et al. "A survey on techniques of secure live migration of virtual machines", *International Journal of Computer Applications, Volume 39-No. 12, February 2012*.
- [10] Michael R. Hines et al. "Post-copy based virtual machine migration using adaptive pre-paging and dynamic self-ballooning", *ACM SIGOPS Operating System Review, Volume 43, Issue 3, July 2009*.
- [11] Fei Ma et al. "Live virtual machine migration using improved pre-copy approach", *Software Engineering and Service Science (ICSESS) 2010 IEEE International Conference*, pp: 230-233.
- [12] Ei Phyu Zaw et al. "Improved live VM migration using LRU and splay tree algorithm", *International Journal of Computer Science and Telecommunications, Volume 3, Issue 3*.
- [13] Hai Jin et al. "Live Virtual Machine Migration with Adaptive Memory Compression", *Cluster Computing and Workshops, 2009, IEEE International Conference*. Pp: 1-10.
- [14] Rakhi K Raj et al. "Live Virtual Machine Migration Techniques- A Survey", *International Journal of Engineering Research and Technology, Volume 1, Issue 7, September 2012*.