

Live Migration of Virtual Machines in Cloud Environment: A Survey

Lalithabhinaya Mallu* and Ezhilarasie R

School of Computing, SASTRA University, Thanjavur - 613401, Tamil Nadu, India;
abhi.cinderella@gmail.com, ezhil@cse.sastra.edu

Abstract

Virtualization is a useful tool for administrators of data centers and clusters. Migration process of VMs mitigates the overloaded condition of data centers and provides the uninterrupted services. The main objective of migration process is to achieve less down time and migration time. Various migration techniques like pre-copy, post-copy and suspend-copy are available to perform live migration of VMs. Pre-copy migration method has optimum performance compare to the other techniques. A great deal of research and study on pre-copy methods has been proposed by researchers. This paper gives a comprehensive survey investigation of different pre-copy live migration methods based on their working principles along with pros and cons.

Keywords: Cloud Computing, Live Migration, Pre-Copy, Virtualization, Virtual Machine (VM)

1. Introduction

Cloud computing is the new way to provide on-demand virtualized resources as services through the internet as pay for use basis¹. Virtualization technology is one of the key factor in the area of cloud computing. It provides the better utilization of computing resources, improves scalability, reliability and availability of resources. The resources such as CPU, memory and I/O are abstracted by virtualization through the creation of VM².

Virtualization allows multiple instance creation on top of single host such that multiple applications can run independently. It separates the operating system from the underlying physical hardware using hypervisor³. Each VM is having its own operating system called guest operating system. Migration of the operating system instances from one physical host to another is an important feature of virtualization.

Virtualization provides maximum profit to the cloud providers with less maintenance cost. Virtual Machine (VM) is an instance which takes network, memory and

I/O resources based on the user request⁴. VMs can be created based on the user request by cloud provider at datacenter.

The process of migrating the VM from source to destination machine without interrupting the other VMs can be called as live migration of VMs. Figure 1 depicts the process of migration between the hosts. Virtual machines can be provisioned on-demand, replicated and migrated. The dynamic load balancing of virtualized hardware is allowed by live migration. The applications running on the virtual machines are confined from the guest operating system⁵. The live migration process is useful when datacenters are facing problems like hotspots, cold spots and load imbalancing. The hotspot is the overloaded condition of a PM, where the performance of a system falls below the minimum acceptance level. The low utilization of physical machine can be defined as cold spot situation and load imbalance can occur when some machines are heavily loaded and some are lightly loaded in a datacenter.

The live migration performance depends on two parameters, Total migration time and down time. Down

*Author for correspondence

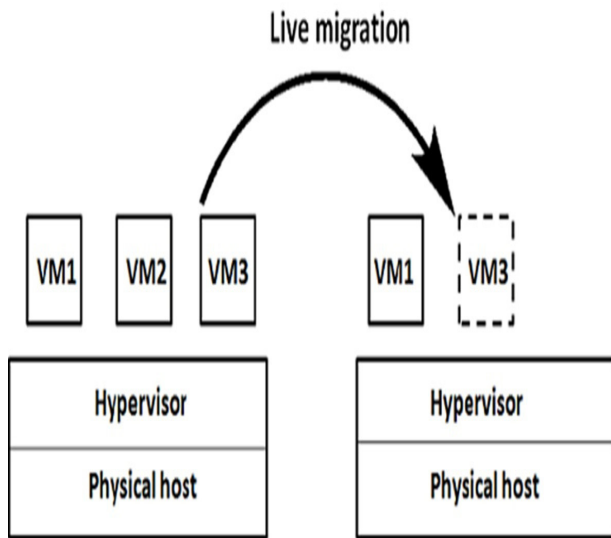


Figure 1. Migration of virtual machines.

time is the period of unresponsive time of VM while migration. The time taken to perform complete migration of VM including downtime is called as total migration time. The key challenge of live migration process⁶ is to achieve an impressive performance with minimum down time and total migration time. The cloud provider gets

maximum profit when the data storage cost and migration cost is less⁷. Cloud storage is scalable in nature and the storage capacity is proportional to the price policy¹⁶. The rest of the paper is arranged as follows: chapter 2 describes the types of migration techniques. In chapter 3 techniques of modified pre-copy are discussed. Chapter 4 concludes the paper.

2. Virtual Machine Migration Techniques

VM migration techniques are mainly classified into two types: non-live and live migration techniques.

2.1 Non-Live Migration Technique

Suspend and copy migration is a technique where the VM stops working on its source and execution state along with memory are migrated to the target machine. Once all the memory pages transferred, VM activates on the target host. VM is in deactivating state until all pages transferred to the destination. So, the downtime and total migration time is equal. Figure 2 shows the steps required to perform stop and copy method. Steps 1 to 3 in the figure represents time of both total migration and down time. Down time is more in this method due to the unavailable services to the user during migration.

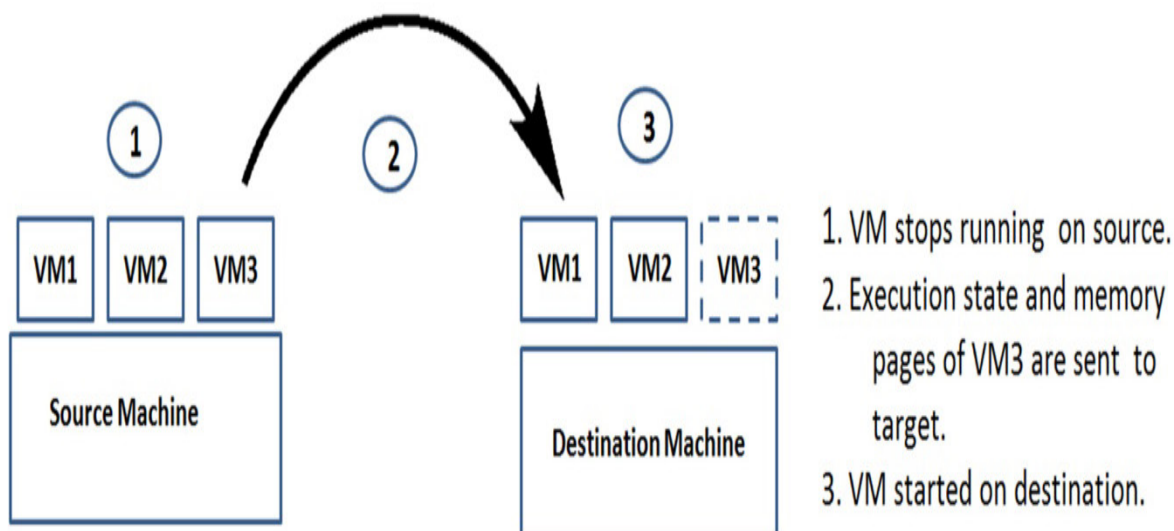


Figure 2. Process of suspend and copy.

2.2 Live Migration Techniques

Live migration techniques are characterized into two types based on the process of memory copy. They are post-copy and pre-copy techniques.

2.2.1 Post-Copy Live Migration

In the post copy approach^{8,9} VM suspends on source and its execution state (CPU, register and Memory pages needed to activate VM on destination) is transferred to destination and VM begin running on destination even the whole pages have not been copied. Page faults can occur when the memory page of the VM is not available at target host, then the corresponding page is transferred from source to destination. Figure 3 demonstrates the working methodology of post-copy migration. The drawback of this method is the occurrence of page faults at the target host.

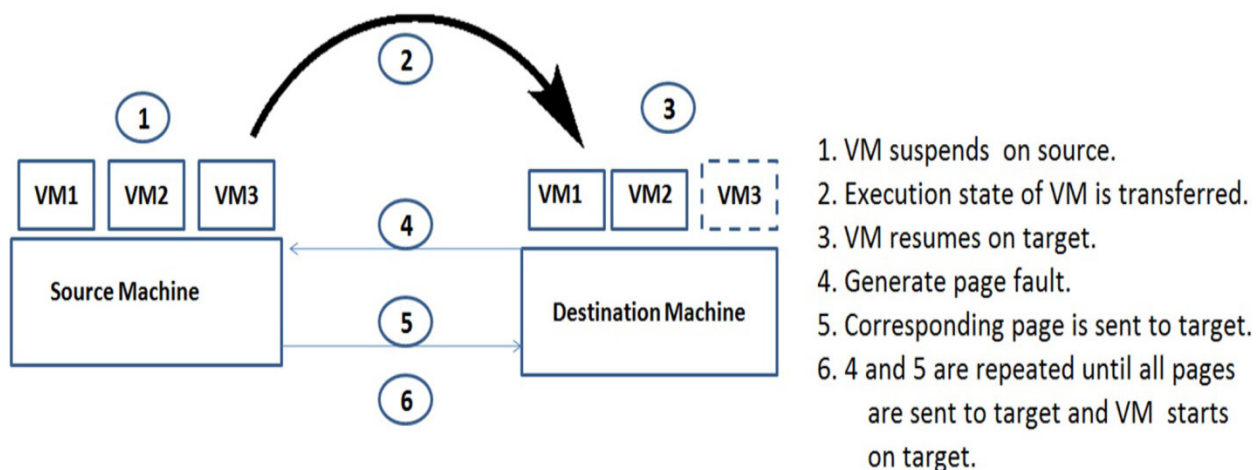


Figure 3. The process of post-copy migration.

2.2.2 Pre-Copy Live Migration

Pre-copy is a live migration technique used by hypervisors like Xen, KVM and VMware. It sends the entire VM to target during first iteration then iteratively transfers the modified memory pages. This process continues until sufficient memory pages copied to target host. In each iteration modified pages can be transferred between the two hosts. In Figure 4 the total migration time includes all the steps whereas down time can be taken by step 3 to 5. In a standard pre-copy approach¹⁰ dirty data generation rate is greater than the memory page transfer because it does not provide a facility to

record the frequently updated memory pages. In case of destination failures, pre-copy method is recommendable than post-copy approach. VM recovery is possible in pre-copy method because the latest copy of memory information is kept safe on source machine including CPU state.

3. Techniques of Modified Pre-copy Migration

This section discusses the several modified pre-copy migration techniques.

3.1 Improved Pre-Copy Approach

Traditional pre-copy method has been modified as improved pre-copy based approach by Ma et al⁸. It facil-

itates the records to keep track of frequently modified memory pages. To-send, To-Skip, To-fix is the three bit-maps used by this approach. The pages modified in the previous iteration can be represented by To-send and the present iteration modified pages can be denoted by To-skip. To-fix⁸ holds the number of fixed pages which are sent in the previous round. The frequently modified previous round pages can be stored by To-send-last extra bitmap and this notation is shown in Table 1.

The unnecessary transfer of frequently updated pages can be minimized in this approach so the number of iterations is reduced. Frequently modified pages can

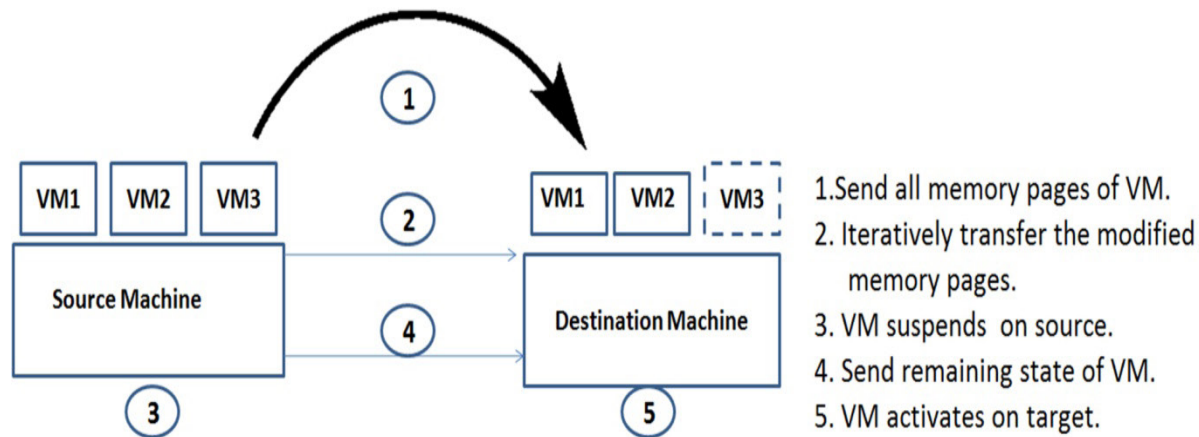


Figure 4. The process of pre-copy method.

Table 1. Bitmap representation in improved pre-copy approach

To-send	0	1	1	0
To-skip	0	1	0	1
Send or not	No	No	Yes	No

be determined based on only To-send and To-skip bitmaps. It is not useful for alternative modified pages in the memory during migration.

3.2 Matrix Bitmap Algorithm

Matrix bit map algorithm overcomes the drawback of improved pre-copy based approach. Many bitmaps¹² are used to determine the memory page transfer between the source and destination. The number of bitmaps collected at each iteration can be determined by a variable called MAP_LEN.

Dirty bitmap collected MAP_LEN times

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 1 \\ 0 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \begin{matrix} P_1 \\ P_2 \\ \vdots \\ P_N \end{matrix}$$

In each iteration unmodified pages can be determined by zero and modified page determined by one. Page weight is calculated based on the bitmap value to compare with the threshold value. The page with lesser weight than the threshold can be sent in this iteration.

The threshold is calculated by 2^{MAP_LEN-1} and if the weight is lesser than the threshold then it can be sent to the destination. MAP_LEN is used to determine the threshold value so, the success of this algorithm depends on this variable. The threshold value must be optimal for producing better outcomes than the above pre-copy method.

3.3 Time Series based Pre-Copy Approach

The Time series approach is proposed by Hu B, et al.¹¹ which enhance the performance of standard pre-copy technique. Standard method uses two bitmaps to determine the condition of modified memory pages. If To-send and To-skip bits are set as one and zero then the memory page transfers between the hosts. Time series approach adds a time series array of historical bitmaps¹¹ named as To-send-h of size N. The pages of history can be recorded by To-send-h array. The value of last round To-send bitmap can be stored by to-send-h array. High dirty pages and low dirty pages are two types of pages. The high dirty page satisfies the below equation and low dirty pages cannot satisfy the equation.

$$\sum_{i=1}^N (p \in \text{To_send}_{h[i]}) \geq K \dots\dots (1)$$

Here size of array denotes N and the series of bit-maps denoted by 'To-send-h' array. The threshold value is denoted as K. The page p modification in iteration can be represented by to-send-h[i] value. High dirty page has more occurrences than the threshold and this value is sent in the previous iteration.

The threshold value is the success factor of this approach. The type of workload decides the threshold value. If the value of the threshold is high then unnecessarily highly modified pages can be transferred to destination.

3.4 Two-Phase Strategy

The approach given by Ma et al.⁸ is one phase strategy which sends the pages depends on previous iteration flag not based on the current iteration. But the two-phase strategy¹³ identify high dirty pages by giving them Second Chance (SC) to the page. In the first phase To-send and To-skip are verified. If both the values are equal to one and zero respectively then it passes to second chance. If the page is clean in this phase then it is migrated to the destination. The duplication of frequently updated pages is avoided by two phase strategy. The steps to follow this methodology are: Initially migration process activates based on the strategy of second chance. If the page is clean for both the iterations transmit the dirty page to target machine. The end condition checks the number of iterations or compares the count of duplicated pages to switch to one phase strategy.

SC strategy can be performed in 28 iterations even though the optimum number of iterations for SC strategy lies in between 1 to 29. After this SC strategy switched to one phase and performs stop and copy. If dirty page count is smaller than 55 it is switched to the one phase because the dirty page number is small. Third condition verifies the dirty page count with memory pages of 2 and half of size of VM. If condition satisfies then it switches to OP strategy.

3.5 Pre-Copy using Memory Compression

Memory compression provides fast and stable VM migration, proposed by jin H et al⁶. Characteristic Based Compression (CBC) algorithm is used to compress the VM memory pages before sending to the

destination. This is one of the techniques to reduce the down time and total migration time. A good compression algorithm gives a higher compression ratio and low overhead. The amount of compressed data is denoted by the compression ratio. The time taken to compress the memory pages by CPU is called overhead. High compression ratio and less overhead are the key parameters for memory compression algorithm⁶. The data is classified into three groups based on this algorithm.

1. The memory page contains both zero and non zero bytes. The value and the non-zero bytes are transfer to the target when the zero byte count is greater than the zero byte limit.
2. The compression of page will take less time when the page contains similar words greater than the threshold of similar words.
3. Otherwise use the algorithm to compress the page with high compression ratio.

Compression and decompression of a memory is overhead in this approach. Unused pages present in the memory are also compressed and sent to destination. The migration time and overheads increases due to the unnecessary transmission of identical pages.

3.6 Memory Ballooning

Memory ballooning is a technique to mitigate the overhead present in the compression algorithm. The unutilized memory pages cannot send to the destination machine so, the time for migration is reduced. A standard method is available in Xen which balloons the unutilized space whenever sufficient space is not available to place a VM. Dynamic Self Ballooning (DSB)⁹ approach always balloons the unutilized pages during its life time. The unused page count is reduced by this method whereas the identical page transmission overhead is still available. The total migration time and downtime are still affected by identical page transmission in this approach.

3.7 WSClock Replacement Algorithm

Working Set (WS) Clock replacement algorithm¹⁴ enhances the performance of standard pre-copy approach by adding the pre processing phase to reduce the transmission data and total migration time. The most recently used memory pages listed in the working set list. Each entry of a page contains 'Time of last use' field. The page which is not frequently modified is

transferred to the destination whereas frequently modified page resides on working set list. This approach mainly depends on working set list. The drawback of this approach is to maintain the working set list properly along with the time.

3.8 Live Migration using LRU and Splay Tree

Prediction working set algorithm¹⁵ combined with LRU (Least Recently Used) cache and splay tree improves the performance of pre copy technique. A working set is defined as the collection of memory pages used in the same process can be denoted as a splay tree. Each

Page can be placed on LRU Cache and its splay tree is constructed with process IDs. If the LRU cache is full, the last page is removed from the cache and defined as working set. It predicts the collection of memory pages in future use to reduce the number of rounds of copying pages. The performance bottleneck of this algorithm is prediction time.

4. Summary

The below Table 2 represents the overview of research done and gives the gaps/future work in their existing work.

Table 1. Summary of live migration techniques

S. No.	Techniques/Algorithms	Tools or/and workloads Used	Future work and/or gaps in existing Technologies
1.	WSClock page replacement algorithm.	Xen	Working set maintenance along with time is the overhead in this approach
2.	Prediction Working set algorithm along with Least Recently Used (LRU) Cache and splay trees.	Cluster composed with six identical servers	Prediction time overhead
3.	Two phase Pre-copy strategy	Simulation program	Improves the algorithm to meet the constraint of a given downtime for a certain class of applications
4.	Improved Time series based live migration technique	Java platform using Net beans	The success factor depends upon the Threshold value.
5.	Matrix Bitmap algorithm	Xen with centos	Establish a Monitor system that supervise the whole system and launch the migration automatically in future.
6.	Improved pre-copy approach by adding Bitmap algorithm	Xen	In future implement this algorithm in wide-area live virtual machine migration.
7.	A novel Memory-compression-based VM (MECOM) migration approach	Xen	Memory compression overheads are present. In future, implement this in the environment where VM holds memory data and huge disk space.
8	Post copy algorithm with adaptive pre-paging approach	Xen 3.2.1	Identical page transmission overhead

5. Conclusion

Virtualization provides flexible and dynamic resource provisioning capabilities to the end users. Live migration of virtual machines transfer the state of the VM from one host to another without interrupting the running VMs. Migration of VMs mitigates the overloaded condition of data centers and maintains the uninterrupted services. The main objective of the migration process is to achieve less down time and migration time. The VM migration includes the transmission of CPU, memory and I/O devices to the target machine. This paper describes various existing enhanced pre-copy based live migration techniques with shared disk space. In future the total migration time has to be still reduced to achieve better performance in data centers.

6. References

1. Furht B, Escalante A. Handbook of cloud computing. Springer; 2010.
2. Clark C, Fraser K, Hand S, Hansen JG, Jul E, Limpach C, et al. Live migration of virtual machines. Proceedings of the 2nd conference on Symposium on Networked Systems Design and Implementation-Volume 2; USENIX Association; 2005.
3. Mandekar AV, Chandramouli K. Centralization of network using openflow protocol. Indian Journal of Science and Technology. 2015; 8(S2):165–70.
4. Pal AS, Pattnaik BPK. Classification of virtualization environment for cloud computing. Indian Journal of Science and Technology. 2013; 6(1):3965–71.
5. Sarma VAK, Rajendra R, Dheepan P, Kumar KS. An optimal ant colony algorithm for efficient VM placement. Indian Journal of Science and Technology. 2015; 8(S2):156–9.
6. Jin H, Deng L, Wu S, Shi X, Pan X. Live virtual machine migration with adaptive, memory compression. IEEE International Conference on Cluster Computing and Workshops (CLUSTER'09); 2009.
7. Rajakumari SB, Nalini C. An efficient cost model for data storage with horizontal layout in the cloud. Indian Journal of Science and Technology. 2014; 7(3S):45–6.
8. Ma F, Liu F, Liu Z. Live virtual machine migration based on improved pre-copy approach. IEEE International Conference on Software Engineering and Service Sciences (ICSESS); 2010.
9. Hines MR, Gopalan K. Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning. Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (ACM); 2009.
10. Liu H, Jin H, Liao X, Hu L, Yu C. Live migration of virtual machine based on full system trace and replay. Proceedings of the 18th ACM international symposium on High performance distributed computing; ACM; 2009.
11. Hu B, Lei Z, Lei Y, Xu D, Li J. A time-series based precopy approach for live migration of virtual machines. IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS); 2011.
12. Cui W, Song M. Live memory migration with matrix bit-map algorithm. IEEE 2nd Symposium on Web Society (SWS); 2010.
13. Lin C-C, Huang Y-C, Jian Z-D. A two-phase iterative pre-copy strategy for live migration of virtual machines. 8th International Conference on Computing Technology and Information Management (ICCM); IEEE; 2012.
14. Sagana C, Geetha M, Suganthe R. Performance enhancement in live migration for cloud computing environments. International Conference on Information Communication and Embedded Systems (ICICES); 2013.
15. Zaw EP, Thein NL. Improved live VM migration using LRU and Splay tree algorithm. Int J Comput Sci Telecomm. 2012; 3(3):1–7.
16. Jeon H, Min Y-G, Seo K-K. A performance measurement framework of Cloud Storage Services. Indian Journal of Science and Technology. 2015; 8(S8):105–11.