



A Survey on Pre-copy Based live Migration of Virtual Machine in Cloud Environment

Sushil Kumar Soni

Computer Technology & Applications
India

Associate prof. Ravi Kant Kapoor

Computer Engineering & Application
India

Abstract— Live migration of virtual machines (VM) has been a strong tool which facilitates system online maintenance, load balancing, power-saving, and fault tolerance, especially in clusters or data centers. Live migration is a necessary feature in today's virtualization technologies. One of the key issues for live migration of virtual machine is how to quickly and transparently migrate virtual machines to meet Service Level Agreements (SLAs). In this paper, we evaluate the time series based approach for live migration of virtual machine. It eliminates the problem of taking threshold as input and combines the second chance (SC) of two-phase strategy to get benefits of both approaches. With the time series based approach, counts the number of zeros and number of ones in array to_send_h, if the possibility of occurrence of ones in array to_send_h is more than zeros then page is high modifying page. After applying modified time series approach, Pages selected to be sent are passed through second chance (SC) strategy, if page is given a second chance and if it is kept clean for second time then it is sent to target.

Keywords— Migration, Virtual Machine, Virtualization, Cloud, Data Center, Pre-Copy, Data-Broker.

I. INTRODUCTION

Virtual machine is basically used [1] in data centers, cluster computing and so on. VM migration is one of the important approaches in the area of physical machine virtualization, which allows application to be transparently migrated along with their execution environments across physical machines [1] [3] [5] [6] [8]. Virtual machine migration is needed for [1] [5] load balancing, server consolidation (power saving) and resource scheduling. For effective migration, downtime and migration time should minimum.

I. Types of Migration:

1. Stop and copy based migration

It is a non live migration technique. Virtual machine completely stop running on source machine and all its memory pages are copied to destination machine [2] [6]. After copying all memory pages, VM is started on destination. Migration time and downtime is same for stop and copy based migration because VM is not started on target host until its all pages are sent to target. Drawback of this method is that VM's services are completely unavailable until it is started on destination causes increased downtime.

Migration time: less number of pages is transferred so time taken by pre-copy iterations will also be reduced.

Down time: Down time is the time when Virtual machine stops running on source and its dirty pages and execution states are transferred and virtual machine starts running on target. Down time may be same or slightly increased as compare to standard pre-copy. It may depend on environment of dirty pages.

Total migration time: Total migration time is the time taken to perform iterations and downtime. Total migration time will be reduced in our approach.

2. Live migration

Live migration of virtual machine allows the VM to be migrated almost without any interrupt to its application's execution [2] [7] [8]. VM migration is an important means for managing applications and resources in large virtualized systems. It enables resource usage to be dynamically balanced in the entire virtualized system across physical host boundaries and it also allows applications to be dynamically relocated to improve performance and reliability.

2.1 Post-Copy Based Migration

In Post-copy, VM stops running on source and only its execution state [1] [9] (CPU, register, memory pages necessary to start VM on target) is sent to target machine and VM starts running on target even the entire memory pages have not been transferred and still resides on source. When VM need any memory page it generates page fault and that corresponding page is sent from source to target machine. When all memory pages are transferred to target machine, VM is completely started on target host. In this method the page fault is one of the overhead.

2.2 Pre-copy Based Migration

Pre-copy method is generally used for live migration. In the first round it transfers [1] all the memory pages to destination machine then iteratively copies pages modified in last round. Process is repeated until the writable working set (WWS) becomes small. When WWS becomes small it performs stop and copy and transfer CPU state and dirty pages. WWS contains the pages modified in each round.

Drawback: In a traditional pre-copy approach [6], dirty pages are iteratively sent until writable working set (WWS) becomes small or a pre defined number of iterations are reached. When those frequently updated pages are transmitted repeatedly, the WWS is not guaranteed to converge [1] across successive iterations, and it will cap the number of copying iterations to the maximum number of iterations.

II. Related Work

I. Pre-copy Based Migration

Pre-copy method is basically used for live migration. In the first round it transfers [1] all the memory pages to destination machine then iteratively copies pages modified in last round. Process is repeated until the writable working set (WWS) becomes small. When WWS becomes small it performs stop and copy and transfer CPU state and dirty pages. WWS contains the pages modified in each round. Pre-copy is more reliable than post copy in case of destination failure [2]. In pre copy the source node still keeps the up to date copy of all memory data of VM and CPU state, so VM can be recovered if necessary, from source but in post copy VM cannot be recovered because destination contains the up to date copy of VM.

Drawback: In a traditional pre-copy approach [6], dirty pages are iteratively sent until writable working set (WWS) becomes small or a pre defined number of iterations are reached. When those frequently updated pages are transmitted repeatedly, the WWS [1] is not guaranteed to converge across successive iterations, and it will cap the number of copying iterations to the maximum number of iterations.

II. Improved Pre-Copy Approach

A modification to traditional pre-copy approach has come in 2010. F. Ma, F. Lui and Z. Lui have given improved pre-copy method for live migration [1]. It is a modification to traditional pre-copy approach described above, since it provides the facility to keep the records of frequently modified pages. This approach [1] uses 3 types of bitmap, to-send, to-skip & to-fix.

to-send - bitmap contains the pages modified in previous iteration.

to-skip - contains the pages modified in current iteration.

to-fix - contains the pages which are fixed to be sent in last iteration.

One more bitmap **to-send-last** is used. to-send-last bitmap contains the pages which are appeared in to-skip bitmap; these pages are frequently modified pages and sent in last iteration.

to-send	0	1	1	0
to-skip	0	0	1	1
Send or not	no	yes	no	no

The improved pre-copy migration procedure as used by Xen [1], includes the following stages in turn: pre-migration, stop and copy, iterative pre-copy, reservation, commitment and activation.

In improved pre-copy approach, those frequently updated pages are put into the to_send_last bitmap page, and transmitted only in the last iteration, so the WWS can converge quickly, and the iterative process can be finished in several iterations. This approach is useful to reduce the no. of iterations by minimizing the unnecessary transfer of frequently updated pages. Describes the improved pre-copy approach.

Pages transferred in iterations: Only the page satisfying both condition is sent to target so reducing the pages transferred in various iterations.

Drawback: Problem with this approach [1] [2] is that it depends only on two bitmap To-send & To skip to identify the frequently updated pages which are useless when the page is modified alternatively, like 101010 then only the last modified copy of page need to be send but it is sent 3 times.

III. Time series based approach

Time series based approach given by Bolin Hu et.al in 2011 [2] enhances the standard pre-copy approach. Standard pre-copy approach identifies the high dirty pages on the basis of only two bitmap to_send and to_skip as described above but a time series based approach uses an array of bitmaps 'to_send_h' of size N [2]. This array is used to record the last N history of pages. If any page is modified in any iteration i then the value of to_send_h is set as 1 for that particular page in iteration i otherwise set as 0. In every iteration time series array is checked, if any page is modified more number of times than the specified threshold K, page is declared as high dirty page for that iteration and to be sent in last round. [2]

Performance of this algorithm depends on parameter threshold K and size of time series array N, for a better result than standard pre-copy method, appropriate ratio of K/N should be chosen. If the appropriate ratio of K/N is not chosen, it cannot give better performance [10].

Working algorithm of time series approach as follows:

```
For every page p;  
IF (to-send=1 & to-skip=1 OR  
to-send=0 & to-skip=1 OR  
to-send=0 & to-skip=0) THEN  
Don't send page p to destination.  
IF (to-send=1 & to-skip=0) THEN  
Check the history of page p from equation (1)  
to determine whether it is high dirty page or not.  
IF it satisfies equation (1) THEN  
Don't send the page p to destination.  
ELSE  
Send page p to destination.
```

Drawback: Success of Time series based approach depends on the value of threshold [3]. Value of threshold should be decided according to type of workload. Value of threshold should not be so high; otherwise unnecessary high modifying pages will be transferred until reach the threshold value.

IV. Two-phase strategy

Instead of taking historical records of pages like in 'Time series based approach', 'two phase strategy' by Cho-chin Lin et.al. in 2012 [3] identify high dirty pages by giving them second chance (SC) to the page. In first phase value of to_send and to_skip is checked, whenever the value of to_send = 1 and to_skip = 0 for a particular page, page is given a second chance (SC) if it is kept clean for a second time then it is sent to target otherwise declared as high dirty page [3]. First phase follow the second chance (SC) strategy while Second phase follow the normal pre-copy strategy. It does not give second chance (SC) to pages. The number of iterations, number of dirtied pages and number of duplicated pages are checked. [3] When the number of dirtied pages is less than 50 or 28 iterations have been completed, switching is performed from one phase to second phase.

III. Future Work

Proposed algorithm is based on time series based approach for live migration of virtual machine. It eliminates the problem of taking threshold as input and combines the second chance (SC) of two-phase strategy to get benefits of both approaches. Our method eliminates the problem of taking threshold value K. Similar to "Time-series based approach" bitmap to_send and to_skip are used. to_send contains the pages changed in previous iteration. to_skip contains the pages modified in current iteration, an array to_send_h with size N is employed to record the history of page in last N iterations, if any page is changed in particular iteration then value of array to_send_h is set to '1' for a particular page in a particular iteration otherwise set to '0'. Problem with time series based method was that it was dependent on threshold value K, if the appropriate ratio of K/N is not selected, it cannot give good performance. Instead of depending on threshold value K, proposed approach counts the number of zeros and number of ones in array to_send_h, if the possibility of occurrence of ones in array to_send_h is more than zeros then page is high modifying page. After applying modified time series approach, Pages selected to be sent are passed through second chance (SC) strategy, if page is given a second chance and if it is kept clean for second time then it is sent to target.

IV. Conclusion

Proposed algorithm combines the history and future record of pages to identify frequently modifying pages; it is obvious that proposed approach provides more restrictions to avoid the unnecessary transfer of dirty pages in iterations. Only the pages, satisfying equation and clean for two repeated round, are sent to target. Algorithm is suitable for both low dirty page environment and high dirty page environment and avoid taking any threshold value as an input. Effect of proposed algorithm on various migration parameters can be easily analysed.

References

- [1] Fei Ma, Feng Liu, Zhen Liu *† * Network Management Research Centre, Beijing Jiaotong University Engineering Research Center of Network Management Technology for High Speed Railway, Ministry of Education Beijing 100044, China mafei2007@gmail.com
- [2] Bolin Hu¹, Zhou Lei¹, Yu Lei², Dong Xu¹, Jiandun Li¹ ¹School of Computer Engineering and Science Shanghai University, Shanghai, 200072, P. R. China ²Department of Computer Science and Engineering The University of Texas at Arlington, Arlington, TX 76019, USA {hubolincn, leiz}@shu.edu.cn, yleil@cse.uta.edu, {dxu,ljd}@shu.edu.cn.

- [3] Cho-Chin Lin, Yu-Chi Huang and Zong-De Jian Department of Electronic Engineering National Ilan University Yilan,Taiwan Email: cclin@niu.edu.tw
- [4] Michael R. Hines and Kartik Gopalan Computer Science, Binghamton University (State University of New York) {mhines,kartik}@cs.binghamton.edu
- [5] Hai Jin, Li Deng, Song Wu, Xuanhua Shi, Xiaodong Pan *Services Computing Technology and System Lab Cluster and Grid Computing Lab School of Computer Science and Technology Huazhong University of Science and Technology, Wuhan,430074, China* hjin@hust.edu.cn
- [6] Haikun Liu, Hai Jin, Xiaofei Liao, Liting Hu, Chen Yu *Services Computing Technology and System Lab Cluster and Grid Computing Lab School of Computer Science and Technology Huazhong University of Science and Technology, Wuhan, 430074, China* hjin@hust.edu.cn
- [7] Weining Liu School of Computer Chongqing University, Chongqing, 400030, China Email: lwncq@163.com Tao Fan School of Computer Chongqing University, Chongqing, 400030, China Email: fantaocqu@yahoo.cn
- [8] en.wikipedia.org/wiki/Live_migration
- [9] Michael R. Hines and Kartik Gopalan Computer Science, Binghamton University (State University of New York) {mhines,kartik}@cs.binghamton.edu
- [10] Neha Agrawal¹ & R.K. Pateriya² ¹M.Tech Scholar, Dept of Computer Science & Engineering, MANIT , Bhopal, India ²Associate Professor, Dept of Computer Science & Engineering, MANIT , Bhopal, India