International Journal of Engineering & Technology, 7 (4.12) (2018) 6-12

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Review on Different Types of Live VM Migration Methods with Proposed Pre-Copy Approach

Arvind Kumar Bhatia[1], Gursharan Singh[2*]

[1]*M.Tech (Computer Science), Lovely Professional University , Phagwara, Punjab (India)*
[2]*Computer Science Department, Lovely Professional University, Phagwara, Punjab (India)*
**Corresponding author E-mail:gursharan.16967@lpu.co.in*

Abstract

Cloud computing is being considered as the future architecture of IT world. Virtualization creates logical resources from physical resources which are allocated with flexibility to applications. Server virtualization is a technique for the division of the physical machine into many Virtual Machines; every Virtual Machine has the capacity of applications execution similar to physical machine. The capability of Virtual Machine migration i.e. dynamic movement of Virtual Machines between physical machines is achieved by virtualization. Migration techniques differ w.r.t order of state transfer. Pre-copy migration method transfers all pages of memory from source to destination while Virtual Machine is executing on source. Post-copy migration is transfer of memory. content after the transfer of process state. Specially, post-copy migration, first copied the process states to the destination machine. Total time of migration, Total pages transferred and Downtime are important parameters considered during live Virtual Machine migration. Many improved live pre copy Virtual Machine migration techniques tries to decrease all the three above mentioned parameters. Proposed approach also tries to minimize all the three performance parameters.

Keywords: *Precopy Migration , Virtualization, Hypervisor, Virtual Machine.*

1. Introduction

Cloud computing is being considered as the future architecture of IT world. The information is exchanged among the server and client within the cloud. Speed of the network is very vital issue in networking. We define "cloud computing" as the pool of shared resources which can be used on-demand. Because of efficient usage and management of resources, improvement in reliability with decrease in operational costs, virtualization technologies are being greatly used by the industry Virtualization creates logical resources from physical resources which are allocated with flexibility to applications. For example server virtualization is a technique for the division of the physical machine into many Virtual Machines; every Virtual Machine has the capacity of applications execution similar to physical machine. Server virtualization allows flexibility in workload assignment to physical machines with the separation of logical resources from the underlying physical resources. This gives us advantage like permitting workload consolidation on a single physical machine instead of executing on many physical machines. Also the capability of Virtual Machine migration i.e. dynamic movement of Virtual Machines between physical machines is achieved by virtualization. Process migration which migrates an executing process from one machine to another is similar to Virtual Machine. With process migration there is movement of the state from one physical machine to another of a running application process. Because of the difficulties in handling the dependencies between various operating system modules process migration has been very less used in reality. Virtual Machine migration is free from such type of limitations. Because movement of running processes along with complete OS is done in Virtual Machine migration the problem of migration becomes simplified and will be efficiently handled. Coud computing are the group of IT services which are given to customer over network. These services will be provided through third party who is the infrastructure owner. Cloud computing is new paradigm that provides services as per demand at very less cost.

In SaaS, software and related data will be deployed through provider. In PaaS, provider provides software programs for specific tasks and In IaaS, provider provides Virtual Machines as well as storage for improvement in customer's business.Cloud computing has close resemblance with grid computing but they are not same.

Virtualization is one of dominant concepts related to cloud computing .Offering efficiency to user requests is prime reason for the usage of Virtual Machines within servers. Live Virtual Machines (VM) migration is primary highlight of "virtualization" where application executing is moved from one system to another without application disruption. There are numerous reasons for live migration in Cloud Data centre.Because of reduction in operational

and business cost due to virtualization, cloud acceptance world-wide increased.

Virtual Machine migration is prime advantage of virtualization. Advantages of Virtual Machine migration are Reducing Energy Costs and Carbon Emissions, Maintenance, Load Balancing and Server Consolidation. The main advantage of server virtualization will be capability for consolidation of multiple Virtual Machines. Here multiple Virtual Machines pack into fewer numbers of physical machines. The physical machines that don't seem running any Virtual Machine will be turned off and it reduces power consumption.

Migration Techniques involves migration of CPU, memory states, hardware device states of running Virtual Machine between hosts. Migration techniques differ w.r.t order of state transfer. Pre-copy migration method transfers all pages of memory from source to destination while Virtual Machine is executing on source. During this process, pages of memory which will become 'dirty' those pages will again be transferred till rate of re-copied pages is not less than page dirtying rate. Fig 1.1 illustrates the process of Pre-copy Virtual Machine Migration. Post-Copy Virtual Machine migration started with suspension of Virtual Machine at source.

The minimal subset of running Virtual Machine state is shifted to destination. Virtual Machine will then be resumed on destination. The remaining Virtual Machine pages are actively pushed by source to target. This activity is known as pre-paging. if Virtual Machine attempts to access page at target that has not yet been transferred, will result in generation of page-fault. These faults, called network faults, will be trapped at destination and sent to source, which transfers faulted page. Post-copy migration is transfer of memory content after the transfer of process state . Specially, post-copy migration, first copied the process states to the destination machine. This permits the Virtual Machine resumption quickly.

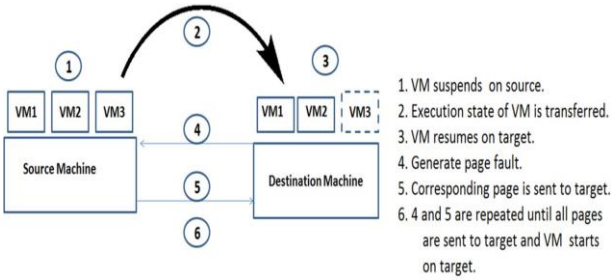

Fig. 1: The process of post-copy migration

Fig 1.1 illustrates the process of post-copy Virtual Machine Migration. Irrespective of the numerous advantages there are some inherent prices that occur due to Virtual Machine migration such as consumption of resources, discontinuity of service, management overhead and vulnerabilities in security. Total time of migration, Total pages transferred and Downtime are important parameters considered during live Virtual Machine migration. Downtime is defined as time during which virtual machine is not in running state.

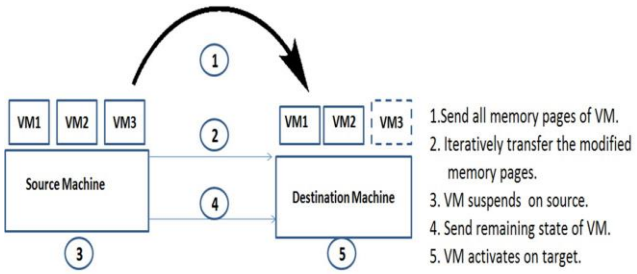

Fig. 2: The process of pre-copy method

Total migration time will be taken as the *time* from when the *migration process* starts from the source machine till the *time* when the destination VM has the control and the source machine can be deleted. Many improved live pre copy Virtual Machine migration techniques tries to decrease all the three above mentioned parameters. Many techniques are described below.

2. Literature Review

Liu et al. (2010) provides algorithm based on slowdown scheduling. In this algorithm they have applied an approach to decrease the dirty rate with the adjustment of resources of CPU allotted to domain of migration, which will result in reduction of activity of CPU, and as a result reduction of rate of dirty pages. This has brought improvement in the performance of migration but it has also an affects on the performance of application executing in migration domain.[1].

In the Micro wiper method proposed by Yuyang Du Hongliang, propagation of memory have been done proficiently during live migration. Micro wiper relies on 2 strategies: One is ordered propagation of memory, second is transfer throttle. Page rewriting rates vary significantly within Virtual Machine memory. Pages having higher rewriting rates, the possibilities of their rewriting are terribly high. If we send these pages they will be likely to become dirtied again and then we have to retransfer them. On the other hand, the pages having less rewriting rates will be best candidates for transfer as per priority. Rather than counting rewriting rate of each page, they have created groups of pages. Group of Contiguous pages of Virtual Machine memory is called stripe. We will get many stripes with equal length after dividing Virtual Machine memory. There has an advantage of stable stripe rewriting rate by reorganizing memory of Virtual Machine into sequence and very decrease in overhead compared with single page.

The size of stripe will be driven as per the size of Virtual Machine memory. They have a straightforward technique for sampling the memory stripes rewriting rates. Micro wiper do the counting of pages dirtied for each stripe for that iteration whenever iteration is completed, the division of this number is then done with the duration of iteration time. This will give the stripe rewriting rate. In every iteration, they have calculated memory stripes rewriting rates, arrange stripes in increasing order.
Then pages dirtied in the stripes are transferred one after another in turn per that order.In every iteration, they initially done calculation of memory stripes rewriting rates from previous iteration, then the memory stripes are arranged as per rewriting rates; They shifted dirty pages of stripes as per that order and dynamically bandwidth of network is estimated; the summation of rewriting rates of stripes transferred will be done and comparison with network bandwidth estimation is done; next iteration will be started instantly when

rewriting rate calculated after accumulation is greater than estimated bandwidth,[2]

Pre-copy algorithm transfers memory pages repetitively. The meaning of iterative is that there are rounds in pre-copying, that is the pages that were shifted in round n were those which are updated in round n-1. More frequency of modification pages have, greater the probability those will be sent. Therefore overall transmission amount has growing. Total time of migration and burden on network would additionally be substantially effected specifically whilst Virtual machines having excessive memory loads. Even though to_skip bitmap will be used for the identification of the pages having frequent changes, this is only being determined during short time from the finish of previous iteration round upto start of next iteration round. If the judgment of page changing frequently has been done in last round, but not during judgment of to_skip in present iteration then it would required be sent yet again. There is big possibility that these same pages will continue to change in present iteration. So pre-copy technique is not performing good under situations when page change frequency is high. Because of these drawbacks, suggest delayed transfer of dirty pages. This algorithm does addition of new bitmap to_delay for having marked pages that were altered both in last iteration as well as current iteration of beginning. These were the pages which are required to be shifted later. So, four bitmap types will be used in this algorithm, to_skip, to_send, to_fix, to_delay [3].

Without considering whether page has been allocated or not, Xen hypervisor migrates full source memory of Virtual Machine to destination. Total time of migration will be directly proportional to size of Virtual Machine memory without workload. Thus, in Memory Explorer module, the authors Yanqing Ma, Hongbo Wang, Jiankang Dong, Yangyang li, Shiduan Cheng(2012) gone through the full Virtual Machine memory with the use of Linux Buddy System which records metadata regarding each unallocated page. Afterwards translation of address of metadata to address of real page will be done. Then, they got page numbers of complete Virtual Machine memory. Accordingly bitmap bits will be set. Every time Migration Daemon before the pre-copy iteration requests bitmap from Memory Explorer. As per bitmap, 1 byte NODATA will be send by Migration Daemon for the pages which are unallocated to destination daemon. Migration daemon will encode allotted pages with the help of RLE algorithm. In case of cost-efficiency of encoding, 1 byte COMPRESSED will be send by migration daemon in order migration daemon of destination could decode compressed data. If not, it will send one byte NOCHANGE for destination.[4]

Alamdari and Zamanifar (2012) provided an approach which is prediction based and this approach is called reuse distance. It does tracking of pages which are frequently modified. It keeps frequently modified pages till last iteration so that there is reduction in retransfer of same pages. This is an approach which applies reuse distance concept for tracking pages that are frequently updated. Based on reuse distance, decision of whether to transfer dirty pages will be taken in each of the phase. The calculation of page Reuse distance is discrete pages updation between contiguous same page updates and for this reuse distance bitmap is maintained. Pages having small reuse-distance will be identified as frequently updated pages. Therefore same page transfer iteratively will be reduced by the application of this method hence result is reduction in no of pages transfers, total migration time and also downtime.[5]

Neha Agrawal & R.K. Pateriya(2013) presents 'two phase strategy' for finding pages having high dirty rates with the help of giving pages second chance. If in first phase, to_send =1 and to_skip=0 for a particular page, then for that page second chance will be considered. In case this page has not been modified in second chance then it will be transferred to destination. if not it will be considered page with high dirty rate. First phase will be as per second chance strategy and Second phase will be as per approach of normal pre-copy. Number of iterations required, number of pages dirtied, numbers of pages duplicated are checked. In case number of pages dirtied left out are < 55 or duplicated number of pages crosses threshold which is predefined or 28 iterations completed, switching will be done from first phase to 2nd phase. Working is as follows:

First Phase:

1. Migration will be as per Second chance approach.
2. if Dirty page is not modified for 2 continuous iterations only then page is Send to destination. Switching condition from first phase to second phase: there is completion of 28 iterations OR number of pages dirtied left out are < 55 OR number. of pages duplicated crosses 2 and half virtual machine size.

Second Phase:

Those pages transferred having to_send=1 and to_skip=0 [6]. In their paper Bubai Das Kunal Kumar Mandal Suvrojit Das (2015) there is source machine memory pages division in fixed size groups which is called block. Size of block will be as per overall pages a machine has. There will be array for every page, the size of which is dependent on threshold value which is number of maximum iterations a machine will have .State of page during every iteration will be stored in this array.1 means page updated , 0 means page not updated.

This array's first element will store an iteration number when a page was last transferred to destination. The sum of pages got dirtied during each iteration will be counted for every block with the help of number of 1's a block have. The arrangement of blocks would done as per dirty counts till last iteration .Block with minimum number of 1(least dirty count) will be selected. Now for that block with least dirty count, current iteration array of pages having 0 are checked. Pages with count of 0s >= number of 1s in array (pages are not modifying so frequently) will required to be shifted to destination. This will be checked from next value when it was last sent till last iteration. After the page has been transferred, first element will now have iteration number when it was sent. This will avoid resending of page if it was not modified till that iteration in which it was transferred. If block has no page which fulfills condition then block having second least count of dirty counts is selected. During the middle round of iteration, only low dirty probability pages (pages are not modifying so frequently) were sent.

Yi Zhong (2014) focuses on optimization of memory state transfer in the iterative copy stage of the pre-copy approach. Our approach uses historical dirty page information periodically collected from the starting of migration at a fixed time interval to make the hot dirty pages recognition more accurate. Further a weighted dirty page rate calculation method is proposed to determine whether a dirty page need be transferred to the target in the current iteration. Experiment results show the optimized approach is better than the existing pre-copy based approaches in terms of downtime, total migration time and the number of transferred pages. They uses a combinatory of the free memory pages elimination based method and the historical data of dirty pages based method.
They first divide types of memory pages into four categories including free memory pages, read-only ones, hot written pages and written pages. This is an enhancement of optimized memory state transferring during the iterations to pre-copy, which introduces the

adaptive and dynamic fine-tuned hot/warm dirty page rate calculation algorithm based on more accurate hot dirty pages sampling. Experiments proved the effectiveness and advantages of their solution.

Table 1: Comparison of VM Migration Techniques

x	Paper	Authors	Year	Methodology	Outcome
1.	Xen live migration with slowdown scheduling algorithm.	Liu Z, Qu W, Liu W, Li K	2010	It decreases the dirty rate with the adjustment of resources of CPU allotted to domain of migration,	improvement in the performance of migration but it has also an affects on the performance of application executing in migration domain.
2.	Microwiper: Efficient Memory Propagation in Live Migration of Virtual Machines	Yuyang Du Hongliang Yu Guangyu Shi Jian Chen Weimin Zheng	2010	Micro wiper relies on 2 strategies: One is ordered propagation of memory, second is transfer throttle.	The experimental results show that Microwiper can significantly reduce downtimeand transferred pages by more than 50%.
3.	Live Migration of Virtual Machines Based on DPDT	Danwei Chen,_, Hanbing Yang, Qinghan Xue, and Yong Zhou	2012	This algorithm does addition of new bitmap to_delay for having marked pages that were altered both in last iteration as well as current iteration of beginning. These were the pages which are required to be shifted later.	The dirty page delay transfer algorithm greatly reduces the amount of the dirty pages which are frequently changed in the transmission.
4.	ME2: Efficient Live Migration of Virtual Machine With Memory Exploration and Encoding.	Yanqing Ma, Hongbo Wang, Jiankang Dong,	2012	A novel approach Memory Exploration and/Encoding (ME2), which first identifies useful pages and then utilizes Run Length Encode algorithm to quickly encode memory.	It efficiently decreas e the total transferred data, total migration time and downtime.
5.	A Reuse Distance Based Precopy Approach to Improve Live Migration of Virtual Machines	Alamdari and Zamanifar	2012	A reuse distance of a VM's page as the number of distinctive pages between two consecutive updating of the same page. using the reuse-distance algorithm, when a page is updated in round n, it will be sent to destination host in the next round (i .e. round n+ 1) if its corresponding bit in the to_skip bitmap is not set and its reuse distance is not less than the dirty_count (number of updated pages in previous round)	Experiments show that with increasing workload, reduction in the number of sent pages reaches 68% without increasing systems' overhead during live migration.
6.	Enhanced time series based pre-copy method for live migration of virtual machine.	Neha Agrawal & R.K. Pateriya	2013	It uses 'two phase strategy' for finding pages having high dirty rates with the help of giving pages second chance. In case this page has not been modified in second chance then it will be transferred to destination.	Total number of pages transferred will be reduced; will result in decreased migration time but downtime may be slightly increased as compare to standard pre-copy technique
7.	Improving total migration time in Live virtual machine migration.	Bubai Das Kunal Kumar Mandal Suvrojit Das	2015	There is source machine memory pages division in fixed size groups which is called block. that block with least dirty count, current iteration array of pages having 0 are checked. Pages with count of 0s >= number of 1s in array (pages are not modifying so frequently) will required to be shifted to destination. This will be checked from next value when it was last sent till last iteration.	During the middle round of iteration, only low dirty probability pages (pages are not modifying so frequently) were sent. So, there will be decrease in total number of pages which will be shifted to destination in middle rounds.

Proposed Method

Proposed methodology working in following three phases:

First Phase: It is concerned with the shifting of pages of memory during first iteration of migration. It is dependent on following conditions:

i. Similar to pre copy method all memory pages will be shifted if none of pages of memory has been modified. (That is virtually an extraordinary case).
ii. Only pages that pass the test of historical statistics described in second phase will be transferred if all of pages of memory has been modified. (That is also virtually an extraordinary case).
iii. Only unmodified pages of memory will be transferred when there is fraction of pages of memory modification.

Because, large number of pages are modified under write intensive type of workload, few pages are left unmodified and therefore, reduction in quantity of pages shifted during first phase.

Second Phase:

i. Threshold value will be calculated to decide the pages which will be considered for transfer. Pages with value less than threshold will be passed to second stage.
ii. History of last three iterations will be checked for the pages passed from the first stage.

One array will be maintained which stores no. of times each page is modified since vm has started execution. During the second phase of Pre-copy we will make use of this array to take decision regarding which pages can be considered for transfer. We will take the average of all the values stored in the array (the counters stored in each location of the array tells how many times particular pages had been modified.)If T represents threshold and

updation is the name of array with size equal to n (n represents the total number of pages in VM) then

T=Summation of all values of updation array/n (1)

Now all the pages having value in updation array less than the calculated threshold value will be considered for transfer but they are not immediately transferred .These pages are required to go through the second stage. In this stage, the pages having updation value less than threshold are now being checked in history record. One array will be maintained which will store the history record of all the pages for the last 3 iterations.

If a particular page has not been modified during the last 3 iterations, that page will be migrated to target machine. As this approach reduces the quantity of pages to be shifted during each iteration, so the total migration time expected to decrease. Figure 3 shows flowchart of proposed VM Migration approach.

Third Phase: In this phase, the last iteration will be completed by sending all the remaining pages to the destination.

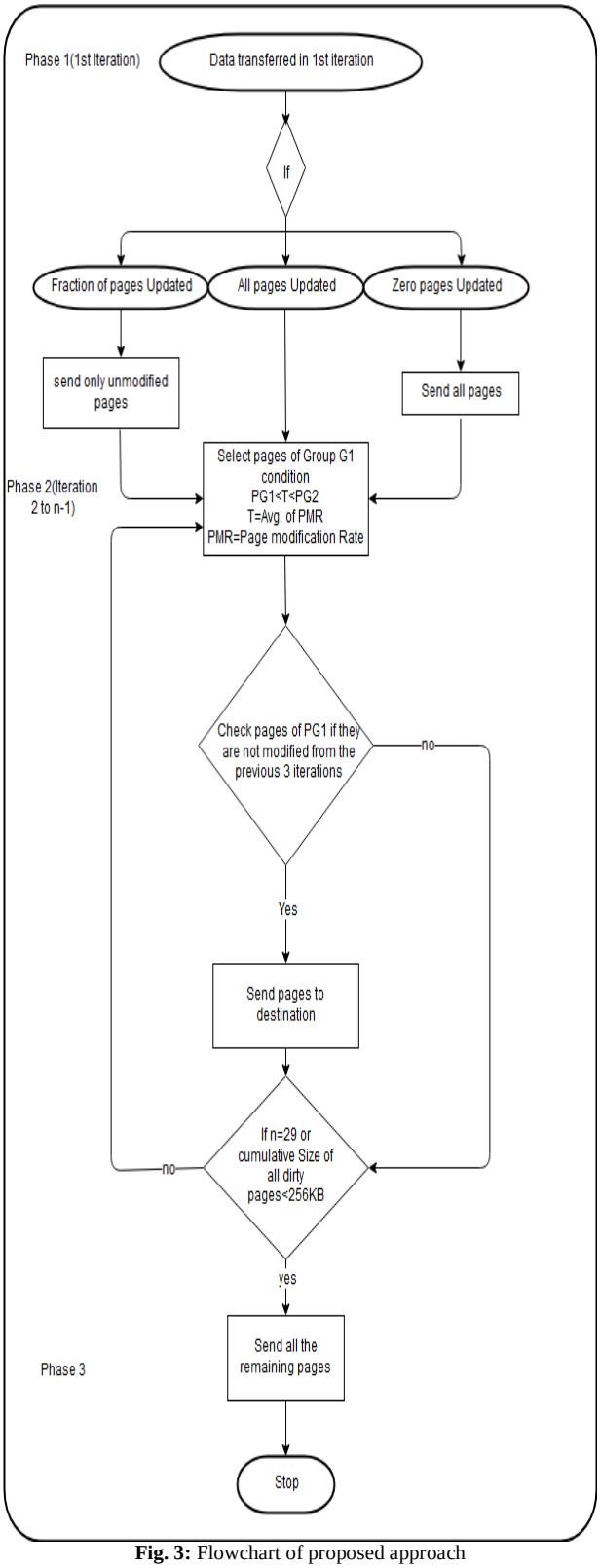

Fig. 3: Flowchart of proposed approach

3. Conclusion

The majority of hypervisors use Pre-copy method for live VM migration. This method is comprised of initial full memory transfer during first phase of migration, iterative transfer of dirty

pages created during the second phase and during the last phase all pages remaining will be transferred to target machine. Problem with this method is the repeated transfer of dirty pages (same pages are transferred multiple times) during the second phase.The downtime and total migration time are increased because of this. Proposed approach first calculates the threshold from the average of number of modifications of all pages. Only pages having modifications lesser than the threshold will be considered for transfer and it does not immediately transfers the page after it satisfies the first condition of average value but considers the last multiple iterations to decide about the transfer of page . This strategy expects to decrease number of dirty pages shifted to destination during iterative part of migration. So it will decrease downtime and total migration time.

References

[1] Liu Z, Qu W, Liu W, Li K : Xen live migration with slowdown scheduling algorithm. In: 2010 International conference on parallel and distributed computing, applications and technologies (PDCAT). IEEE, pp 215–221

[2] Yuyang Du Hongliang Yu Guangyu Shi Jian Chen Weimin Zheng: Microwiper: Efficient Memory Propagation in Live Migration of Virtual Machines.2010 39th International Conference on Parallel Processing

[3] Danwei Chen,_, Hanbing Yang, Qinghan Xue, and Yong Zhou: Live Migration of Virtual Machines Based on DPDT. China conference on wireless sensor networks CWSN 2012 pp26-33

[4] Yanqing Ma, Hongbo Wang, Jiankang Dong, Yangyang li, Shiduan Cheng: ME2: Efficient Live Migration of Virtual Machine With Memory Exploration and Encoding. 2012 IEEE International Conference on Cluster Computing

[5] Alamdari and Zamanifar: A Reuse Distance Based Precopy Approach to Improve Live Migration of Virtual Machines. 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing

[6] Neha Agrawal & R.K. Pateriya : Enhanced time series based pre-copy method for live migration of virtual machine. International Journal of Advances in Engineering & Technology, July 2013. ISSN: 22311963

[7] Bubai Das Kunal Kumar Mandal Suvrojit Das :Improving total migration time in Live virtual machine migration.Proceedings of the sixth International conference on computer and communication Technology (pages 57-61) ICCCT 15 ISBN 978-1-4503-3552-2

[8] Praveen Jain , Ratish Agrawal : An Improved Pre-copy Approach for Transferring the VM Data during the Virtual Machine Migration for the Cloud Environment. I.J. Engineering and Manufacturing, 2016, 6, 51-59 http://www.mecs-press.net/ijem

[9] Fang Yiqiu, Chen Yuyang, Ge Junwei :Improvement of Live Migration mechanism for Virtual Machine based on Pre-copy. 3rd International Conference on Materials Engineering, Manufacturing Technology and Control (ICMEMTC 2016).

[10] Sharma, S., & Chawla, M. (2016). A three phase optimization method for precopy based VM live migration. *SpringerPlus*,*5*(1).

[11] Jain, P., & Agrawal, R. (2016). An Improved Pre-copy Approach for Transferring the VM Data during the Virtual Machine Migration for the Cloud Environment, (November), 51–59.

[12] Balalaie, A. (2016). Microservices Architecture Enables DevOps : An Experience Report on Migration to a Cloud-Native Architecture.The Architectural Concerns for Microservices Migration The Architecture of Backtory Before the Migration. *IEEE Software*, *33*(3), 42–52.

[13] Technology, M. (2016). Improvement of Live Migration mechanism for Virtual Machine based on Pre-copy Fang Yiqiu 1 , Chen Yuyang 1, Ge Junwei 1 1, (Icmemtc), 1725–1729.

[14] Amiri, P. A. D., Rad, S. Z., & Isfahani, F. S. (2016). Providing a solution to improve pre-copy method for migrating virtual machines in cloud infrastructure. *Journal of Theoretical and Applied Information Technology*, *92*(2), 225–234.

[15] Slominski, A., Muthusamy, V., & Khalaf, R. (2015). Building a multi-tenant cloud service from legacy code. *Proceedings – 2015 IEEE International Conference on Cloud Engineering, IC2E 2015*, 394–396.

[16] Devices, S. N., & Xu, X. (2016). A Hypervisor Approach to Enable Live Migration with Passthrough, 15–23. [40]. Jain, L. C., Patnaik, S., & Ichalkaranje, N. (2015). Intelligent computing, communication and devices: Proceedings of ICCD 2014, volume 1. *Advances in Intelligent Systems and Computing*, *308 AISC*(VOLUME 1), 303–319.

[17] Patel, P. D. (2014). Live Virtual Machine Migration Techniques in Cloud Computing : A Survey, *86*(16), 18–21. [45]. Zhong, Y., Xu, J., Li, Q., Zhang, H., & Liu, F. (2014). Memory state transfer optimization for pre-copy based live VM migration. *Proceedings - 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications,WARTIA 2014*, 290–293.

[18] Zhang, Z., Xiao, L., Zhu, M., & Ruan, L. (2014). Mvmotion: A metadata based virtual machine migration in cloud. *Cluster Computing*, *17*(2), 441–452.

[19] Wu, T. Y., Guizani, N., & Huang, J. S. (2017). Live migration improvements by related dirty memory prediction in cloud computing. *Journal of Network and Computer Applications*, *90*, 83–89.

[20] Deshpande, U., & Keahey, K. (2017). Traffic-sensitive Live Migration of Virtual Machines. *Future Generation Computer Systems*, *72*, 118–128.