AN OPTIMIZED MULTISTAGE PRECOPY APPROACH FOR TRANSFERRING VM STATE DURING LIVE MACHINE MIGRATION

A Dissertation-II

Proposal submitted

To

Department of Computer Science and Engineering

In partial fulfilment of the Requirement for the

Award of the Degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGG

by

ARVIND KUMAR BHATIA

Reg. No.41500046

Under the guidance of

GURSHARAN SINGH

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

December 2017

TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering

Program : P172-T::M.Tech. (Computer Science and Engineering) [Part Time]

COURSE CODE :	CSEP548	**REGULAR/BACKLOG :**	Regular	**GROUP NUMBER :**	CSERGD0361
Supervisor Name :	Gursharan Singh	**UID :**	16967	**Designation :**	Assistant Professor
Qualification :	____________________		**Research Experience :**	____________________	

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Arvind Kumar Bhatia	41500046	2015	K1521	8427635577

SPECIALIZATION AREA : Networking and Security **Supervisor Signature:** ________________

PROPOSED TOPIC : Security Considerations During Virtual Machine Migration

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	**Parameter**	**Rating (out of 10)**
1	Project Novelty: Potential of the project to create new knowledge	7.00
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.25
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.50
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	8.25
5	Social Applicability: Project work intends to solve a practical problem.	8.00
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	8.00

PAC Committee Members		
PAC Member 1 Name: Prateek Agrawal	UID: 13714	Recommended (Y/N): Yes
PAC Member 2 Name: Deepak Prashar	UID: 13897	Recommended (Y/N): Yes
PAC Member 3 Name: Raj Karan Singh	UID: 14307	Recommended (Y/N): NA
PAC Member 4 Name: Pushpendra Kumar Pateriya	UID: 14623	Recommended (Y/N): Yes
PAC Member 5 Name: Sawal Tandon	UID: 14770	Recommended (Y/N): NA
PAC Member 6 Name: Aditya Khamparia	UID: 17862	Recommended (Y/N): NA
PAC Member 7 Name: Anupinder Singh	UID: 19385	Recommended (Y/N): Yes
DAA Nominee Name: Kuldeep Kumar Kushwaha	UID: 17118	Recommended (Y/N): NA

Final Topic Approved by PAC: **Security Considerations During Virtual Machine Migration**

Overall Remarks: Approved

PAC CHAIRPERSON Name: 11024::Amandeep Nagpal **Approval Date:** 04 Nov 2017

11/27/2017 4:29:04 PM

ABSTRACT

Virtualization is one of the major features responsible for the acceptance of cloud world-wide as it leads to reduced operational and investment cost. One of the important benefits of Virtualization is Virtual Machine (VM) migration. Live Virtual machine migration technique can be defined as the process of migrating the state of a physical machine to another without the disruption of the application running on the Source VM. There are many benefits of using live VM migration such as Reducing Energy Costs and Carbon Footprint, Facilitating Maintenance, Load Balancing, Server Consolidation. However migration has some associated costs such as Resource Consumption, Service Discontinuity, Management Overhead, Security Vulnerabilities. The main aim of the proposed study is to improve the process of precopy live vm migration i.e to reduce the number of pages transferred during the iterative transfer so that there will be minimum total migration time and downtime.

DECLARATION

I hereby declare that the dissertation-II proposal **AN OPTIMIZED MULTISTAGE PRECOPY APPROACH FOR TRANSFERRING VM STATE DURING LIVE MACHINE MIGRATION** entitled, submitted for M.Tech Degree is entirely my original work and all ideas and references have duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date:_____________ **Investigator**

Reg. No. 41500046

CERTIFICATE

This is certifying that Arvind Kumar Bhatia has completed M.Tech dissertation-II proposal **AN OPTIMIZED MULTISTAGE PRECOPY APPROACH FOR TRANSFERRING VM STATE DURING LIVE MACHINE MIGRATION** titled under my guidance and supervision. To the best knowledge, the present work is the result of her original investigation and study. No part of the dissertation has ever been submitted for any diploma and degree.

The dissertation is fit for the submission and partial fulfilment the conditions for award of the M.Tech Computer Science & Engg.

Signature of Advisor

Gursharan Singh

Date:--------------------------

Counter Signed by:

1) **Concerned HOD:**
   HoD’s Signature: ________________
   HoD Name: ____________________
   Date: ___________________

2) **Neutral Examiners:**

   External Examiner

   Signature: _______________

   Name: ___________________

   Affiliation: ______________

ACKNOWLEDEGEMENT

I express my deep gratitude to my honourable guide Gursharan Singh, Department of Computer Science and Engineering, Lovely Professional University, Phagwara for providing me stimulating guidance, continuous encouragement and support throughout the preparation of thesis.

I would also like to thank all the faculty members, LPU Phagwara and all my friends who were always there at the need of the hour and provided all the help and support, which I required for the completion of the thesis.

Last but not the least; I would like to thank God for not letting me down at the time of crisis and showing me the silver lining in the dark clouds.

ARVIND KUMAR BHATIA

TABLE OF CONTENTS

TABLE OF CONTENTS

List of Figures

CHAPTER-1

INTRODUCTION

1.1 CLOUD COMPUTING

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 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 virtual machines. Also the capability of VM migration i.e. movement of VM dynamically from one physical machine to another is achieved with the help of virtualization. Process migration which migrates an executing process from one machine to another is similar to VM. 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. VM migration is free from such type of limitations. Because movement of running processes along with complete OS is done in VM migration the problem of migration becomes simplified and will be efficiently handled.[19]

1.2CLOUD SERVICES:

Cloud 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 owner of infrastructure. Cloud computing has been considered as new paradigm that provides services as per demand at very less cost. The 3 service models are given below:

1. Software as a service (SaaS)
2. Platform as a service (PaaS)
3. Infrastructure as a service (IaaS).

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.

1.3TYPES OF CLOUDS:

Clouds are of three types:

1.Public cloud

2.Private cloud

3.Hybrid cloud.

With a public cloud, all hardware, software and another infrastructure is managed by the provider. A private cloud consists of computing resources used exclusively by one business or organization. Hybrid clouds combine on-premises infrastructure, or private clouds, with public clouds so organizations can reap the advantages of both. Google, Amazon, IBM provides cloud services.

1.4 VIRTUALIZATION AND LIVE MIGRATION OF VIRTUAL MACHINES:

Virtualization is one of dominant concepts related to cloud computing .Offering efficiency to user requests is prime reason for the usage of VM within servers. Live migration of Virtual Machines (VM) is a 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.

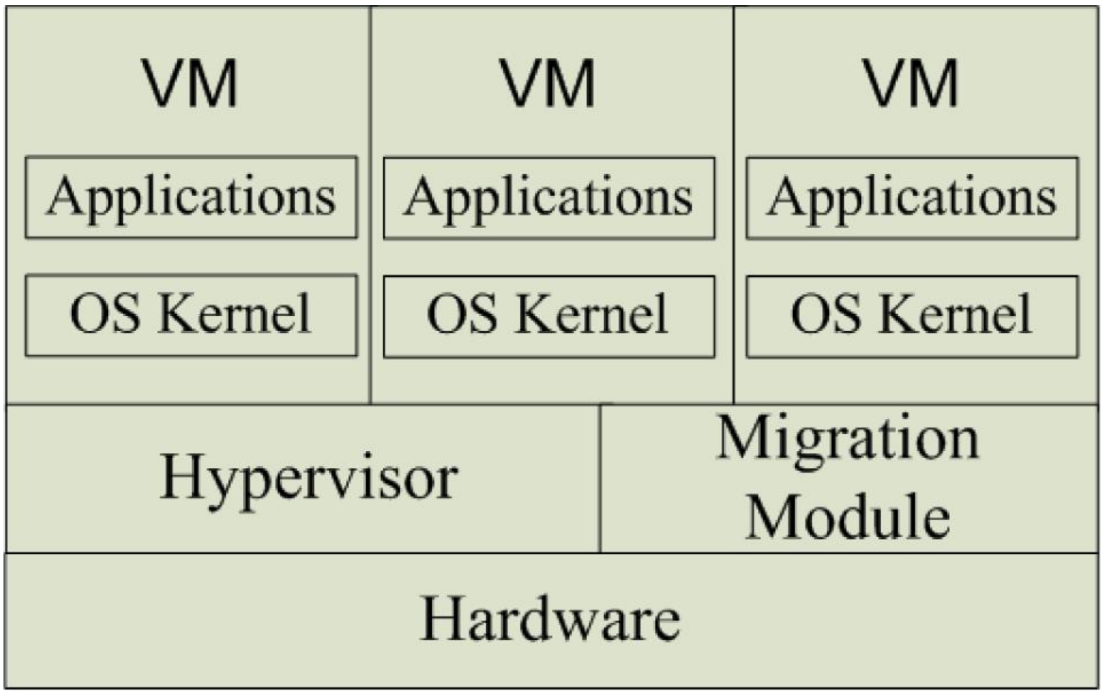


Fig 1: Virtualization of physical host

1.4.1 POPULAR HYPERVISORS

Hypervisors are mainly used for the creation of the virtual machines. There are three main areas where they are used, at server i.e. server virtualization, at storage i.e. storage virtualization and at network i.e. network virtualization.

All the above listed type of virtualization requires hypervisor. The only variation is in details and load that hypervisor could carry for management of every VM.

XEN

· It is oldest available open source virtualization technology used from approximately 5 years.

· Xen uses para-virtualization and hence it runs efficiently without the need of emulation.

· But Xen is very complex in nature .

KVM (Kernel based Virtual Machine)

· It resides in Linux kernel itself.

· Virtual machine can be created by just loading a module in the kernel.

· It has strengths such as security, Memory management, Live Migration, Performance, Scalability, and guest support.

· KVM support is pre-built into fedora Linux kernel for fedora 7 and above release.

· To fully utilize KVM following additional packages are required

· Qemu-KVM

· Virt-Manager

· Virt-Viewer

· Python-virtnist

1.4.2 LIVE MIGRATION

Because of reduction in operational and business cost due to virtualization, cloud acceptance world-wide increased. Virtual Machine migration is prime advantage of virtualization. Live Virtual machine migration is migrating the state of a physical machine to another without the disruption of the application running on the Source VM. There are various Virtual Machine migration techniques, such as:

A. Energy Efficient Migration Technique

The power usage of Data Centre is directly proportional to the number of servers and the cooling Systems. This consumes a lot of power about 70% of maximum power consumption even at low usage. So we need migration methods that save the energy of server by optimum resource utilization.

B. Load Balancing Migration Technique

Load balancing migration method distributes load among servers for the improvement of scalability in cloud environment. This minimizes resource consumption; over provisioning is avoided of resources.

C. Fault tolerant Migration Technique

In the event of system failure, the fault tolerant migration technique keeps running the application. It shifts application from the fail VM to another VM and here future prediction is used.

1.4.3 Advantages of VM migration

1. Reducing Energy Costs and Carbon Emissions

Migration in wide area is being driven by different type of constraints, like energy potency, availableness of renewable resources and electricity price. In response to electricity price fluctuations between completely different regions the workload can be shifted so that cost-effective running of applications is ensured. [19]

2. Maintenance

VM Migration is additionally necessary in relation to the maintenance because additional flexibility is being provided to data center operators like the capability for migration of services in order that operations of routine maintenance can be performed. This can be significantly helpful because maintenance work typically needs human interventions which can induce errors. Live migration reduces these types of risks by permitting VMs migration between different clusters without discontinuation of ongoing services. [19]

3. Load Balancing

Although server consolidation provides several advantages like energy saving and operational cost, it will result in performance degradation. Several VMs packing in less number of physical machines will result in machine overloading, where usage of total resources of VMs crosses machine capability. Due to this, applications performance is going to suffer.[19]

4. Server Consolidation

The main advantage of server virtualization will be capability for consolidation of multiple VMs. Here multiple VMs pack into fewer numbers of physical machines. The physical machines that don’t seem running any VM will be turned off and it reduces power consumption. [19]

1.4.4 Live Migration Techniques

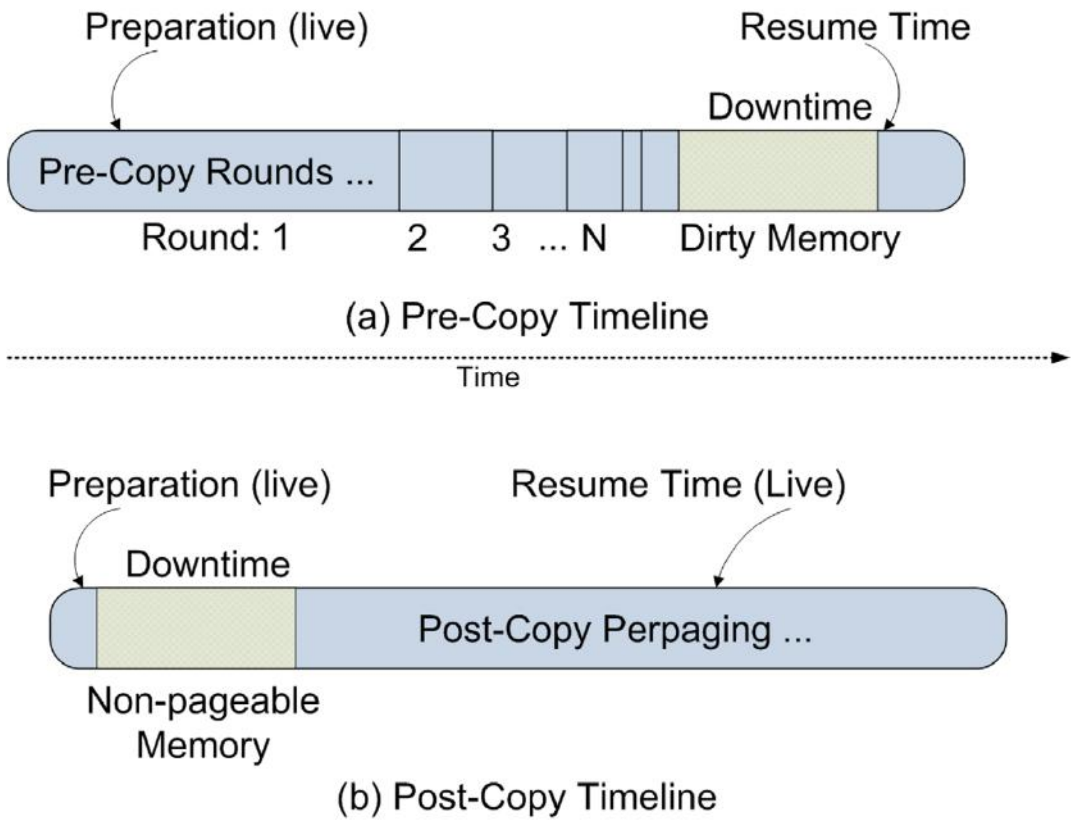
Migration Techniques involves migration of CPU, memory states, hardware device states running VM from one host to another. Migration technique differs in order of state transfer.

1) Pre-copy Migration: In pre-copy memory migration, the Hypervisor copies all the memory pages from source to destination while the VM is still running on the source. If some memory pages change (become 'dirty') during this process, they will be re-copied until the rate of re-copied pages is not less than page dirtying rate.

2) Post Copy:. Migration of VM is started with the suspension of VM at source. The minimal subset running state of Virtual Machine is shifted to target The VM will be resumed then at target. Concurrently, the source actively pushes the remaining memory pages of the VM to the target - an activity known as pre-paging. At the target, if the VM tries to access a page that has not yet been transferred, it generates a page-fault. These faults, known as network faults, are trapped at the target and redirected to the source, which responds with the faulted page.. Post-

copy migration is transfer of memory content after the transfer of process state (Figure 2). Specially, post-copy migration, first copied the process states to the destination machine. This permits the VM resumption quickly.

Figure 2. Timeline for pre-copy vs. post-copy[13]



1.4.5 MIGRATION COSTS

Irrespective of the numerous advantages there are some inherent prices that occur due to VM migration. This section puts light on these costs.

1. Consumption of Resouces

Migration of VMs from one location to a different will consume varied sorts of resources. Varied studies within the literature have given the evidence of the resource overhead due to migration of VM. Some had discovered overhead of CPU due to VM migration upto 20%

utilization of machine, and therefore will not be ignored. Nelson, Lim, and Hutchins (2005) reported that it can need up to 30% CPU utilization to achieve maximum network throughput for VM migration over a gigabit Ethernet link[14]. [19]

2. Discontinuity of service

Inspite of advancement of live-migration technologies, service inaccessibility due to migration remains inescapable. Moreover, the applications executing on source as well as destination machines throughout the migration process will experience degradation in performance because of the extra resource overhead incurred. For example, Voorsluys, Broberg, Venugopal,and Buyya (2009),discovered that the typical web application experienced 3 seconds of downtime and more than 44 seconds of downgraded performance. They also found SLA violations because of VM migration [17] [19]

3. Management Overhead

A VM migration must be done to a physical machine which fulfills new requirements of it. There may be many physical machines that could fulfill the requirements of VM's. But, finding the most effective VM placement must consider many parameters. For example, compliance of security could have the requirement that the VM will be executed on machines with special features of security and trustworthiness. As a result, finding a best placement that creates a balance of objectives is a tedious task. [19]

4. Vulnerabilities in security

Vulnerabilities in security during VM migration can be exploited for attacks. Oberheide et al. (2008) had provided vulnerabilities exposed during live Virtual Machine migration with Xen. Specially, vulnerabilities will occur at 3 levels:

1. **Control plane:** Due to fake migration commands issued by malicious users victim VMs may migrate to locations which are undesirable.

2. **Data plane:** Because there is need of memory transfer and content of disk during VM migration malicious user could manipulate actively the transmitted content which may result in malfunction of VM.

3. **Migration module:** The protection of migration module from users is necessary otherwise they can gain full access of migrating virtual machines. [19]

CHAPTER-2

REVIEW OF LITERATURE

Zhang et al. (2010): They provided compression based method together with data deduplication. With use of the run-time memory image self-similarity, authors have applied RLE method for the migration to get rid of duplicate memory data. Hash based fingerprints have been used to calculate the similarity of pages. For the implementation ,FNHash and FPHash LRU hash tables have been maintained. Migration performance has improved with respect to space, CPU resource overhead.[12]

Liu et al. (2010) provides a slowdown scheduling algorithm. This algorithm has applied an approach to decrease the dirty rate with the adjustment of CPU resources allotted to migration domain. This has resulted in CPU activity reduction, 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.[11]

Yuyang Du Hongliang Yu Guangyu Shi Jian Chen Weimin Zheng(2010) Propagation of memory have been done proficiently in the proposed Microwiper method during live migration. Microwiper relies on 2 strategies: one is ordered propagation of memory, second is transfer throttle. Page rewriting rates vary significantly within VM 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 VM memory is called stripe. We will get many stripes with equal length after dividing VM memory. There has an advantage of stable stripe rewriting rate by reorganizing memory of VM into sequence and very decrease in overhead compared with single page. The size of stripe will be driven as per the size of VM memory. They have a straightforward technique for sampling the memory stripes rewriting rates. Microwiper 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. Lastly, there is application of aordered propagation at the granularity of memory stripe: in every iteration, they have calculated memory stripes rewriting rates, arrange them in increasing order, and then pages dirtied in stripes 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 estimate network bandwidth; the summation of transferred stripes rewriting rates 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,[6]

Danwei Chen,_, Hanbing Yang1, Qinghan Xue, and Yong Zhou(2012)Pre-copy algorithm transfers memory pages repetitively. The meaning of Iterative is that three 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, this paper suggests 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 . The steps of this algorithm will be given below:

After adding the bitmap to_delay, the page transfer decision in every round will be as per table..[5]

to_send	0	0	0	0	1	1	1	1
to_skip	0	0	1	1	0	0	1	1
to_delay	0	1	0	1	0	1	0	1
Transfer or not	No	Yes to_delay=0	No	No	Yes	No	No to_delay=1	No to_delay=1

Yanqing Ma, Hongbo Wang, Jiankang Dong, Yangyang li, Shiduan Cheng(2012) Without considering whether page has been allocated or not, Xen hypervisor migrates full memory of source VM to destination node. Total time of migration will be directly proportional to size of VM memory without workload. Assume there is two- mega-Bytes VM with half memory used while another half free. While migrating such VM, it is certain that one-Giga-bytes of memory transfer will be unnecessary, which will result in a big wastage of bandwidth of network. Xen is not able for providing Guest OS memory utilization. Thus, in Memory Explorer module, the authors gone through the full VM memory with the use of Linux memory management mechanism Buddy System which records every unallocated page's meta-data. Afterwards translation of meta-data's address to real page's address will be done. Then, they got page numbers of complete VM memory. Accordingly bitmap bits will be set. Every time Migration Daemon before the pre-copy iteration would request Memory Explorer for bitmap. As per bitmap, Migration Daemon will send only one byte NODATA for the pages which are unallocated to destination daemon. Migration daemon will encode allotted pages with the help of RLE algorithm. In case encoding is cost-efficient, i.e. size of compressed data is less w.r.t. page size (usually 4KB), migration daemon will send one byte COMPRESSED in order that migration daemon of destination could decode compressed data. if not, it will send one byte NOCHANGE for destination.[9]

Alamdari and Zamanifar (2012) This provides 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 defined as number of discrete pages updation in between two contiguous updates of same page and to do this it maintains the reuse distance bitmap. Pages having small reuse-distance will be identified as frequently updated pages. Therefore transfer of same page iteratively will be reduced with the application of this method hence result is reduction in total number of pages transferred, migration time and downtime as well.[10]

Neha Agrawal & R.K. Pateriya(2013) This paper presents 'two phase strategy' for finding pages having high dirty rates with the help of giving pages second chance. The value of to_send and to_skip is checked during first phase, whenever to_send =1 and to_skip=0 for a particular page, that page will be given second chance. In case this page has not been modified for a second time then it will be sent to target if not it will be considered high dirty page. First phase will follow second chance strategy and Second phase will follow normal pre-copy approach. 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 number of pages duplicated crosses the threshold of virtual machine which is predefined or there is completion of 28 iterations, switching will be performed from one phase to second phase. Working is as follows:

First Phase:

1. Migration will be according to Second chance approach.

2. Dirty page will be Send to destination if page will not be modified for two contiguous iterations.

Switching condition from one phase to second phase: there is completion of 28 iterations OR number of pages dirtied left out are < 55 OR no. of pages duplicated crosses 2 and half of size of virtual machine.

Second phase: transfer of pages having to_send=1,to_skip=0.

The concept used for the identification of pages with high dirty rate in time series based method and second chance (SC) techniqur is different. One of the methods considers high dirty pages based on history and the other considers based on future record by providing a page second chance. Proposed approach is a combination of both methods having slightly modifies time series approach.[2]

Bubai Das Kunal Kumar Mandal Suvrojit Das (2015) In this paper 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, the array of pages having 0 in the current iteration will be checked. Pages with number of 0s greater than or equal to 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 the pages with low dirty probability (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.

When number of iterations crosses the threshold value, source VM is suspended, all left out dirty pages were sent to destination, virtual machine at target is resumed.[4]

Praveen Jain , Ratish Agrawal (2016) There are two phases in this proposed approach. The pages that are updated in last iteration but not modified in current iteration are separated out in first phase. Division of pages is done as per second phase into two types i.e., pages with high dirty rate and normal depending upon the updation in last some iterations. The filtered pages from first phase are checked for the number of times the page modification has been done from history. The page that is modified more times is required not to be sent during current iteration. On the other way send page to destination VM. Simple concept used to decide the transfer of page. In case number of modifications of page is more than unmodifications from the history record, the page will be dirty.

it is implemented in CloudSim simulator for evaluation of performance of the planned approach and comparison with existing pre-copy approach which is based on time series is done in terms of total migration time and down time. The result proves better result for proposed approach.[1]

Fang Yiqiu, Chen Yuyang, Ge Junwei(2016) Pre-copy algorithm get rid of repeated transfer of dirty pages based on probability prediction of memory pages becoming dirty . But, prediction of probability will fail when the dirty pages increase significantly,. This paper proposes an Adaptive Bandwidth technique within pre-copy dependent on probability prediction to solve this problem. This algorithm improves transmission bandwidth for decreasing transmission time when dirty pages rate suddenly increases. Results after doing experiments prove that when dirty page rate increases unexpectedly or is large, that strategy improves transmission bandwidth, decrease the iterative transfer total time as well as downtime, as a result improving performance of migration.[3]

Parvin ahmadi doval amiri, Shayan zamani rad, Faramarz safi isfahani(2016) The crux of their proposed method is that to adopt CPU speed (or CPU Frequency) during migration time. In other words, they reduce the speed of CPU gradually, and as a result the changing rate of memory pages will be reduced ultimately. In order to implement this idea, they write an algorithm that decreases the frequency of virtual machine's CPU (vCPU) at the time of moving, just for a short period of time. This declining in the CPU speed will reduce the performance of selected virtual machine, therefore, a reduction on writing operations and modifying pages will happen consequently. The effectiveness of this solution emerges in such cases that they have write-intensive workloads on the most of VMs. It should be emphasized that this policy,vCPU frequency reduction, will apply till the migration process finishes.

In the proposed algorithm, the vCPU frequency is decreased until the rate of sending pages through the network (network bandwidth) becomes higher than memory page change rate. After a number of steps, migration process will come to the second phase, stop and copy, and virtual machine becomes suspended. In this stage, if the pre-copy method uses, the migration process takes long time and too many rounds, close to 30.Moreover, in the worst case, live migration will turn into the non-live, because of the high changing rate. Page memory changing rate and network bandwidth are compared at first step. If changing rate is higher, the migration process will not finish. So, the algorithm records the status of vCPU. Then, it reduces the vCPU frequency to half. Then it checks whether the reduction in the frequency reduces memory page changing rate or not. If nothing happens, it will reduce the frequency to half again. This loop continues until the memory pages changing rate becomes lower than network bandwidth. Finally, after transferring virtual machine to the destination, the main frequency will be adjusted again and all the states will roll back on the machine.[7]

Sangeeta Sharma and Meenu Chawla (2016) There are three phases in the proposed approach:

First Phase: it is concerned with the shifting of pages of memory during first iteration of migration.

Second Phase: it deals with shifting of pages within iteration ‘2’ till ‘n − 1’.

Third Phase: In this phase comes last iteration.

First Phase: it is dependent on following conditions:

1 Similar to precopy method all memory pages will be shifted if none of pages of memory has been modified. (That is virtually an extraordinary case).

2. Only pages that pass the test of historical statistics described in 2nd phase will be transferred if all of pages of memory has been modified. (That is also virtually an extraordinary case).

3 Only unmodified pages of memory will be transferred when there is fraction of pages of memory modification.

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

Second Phase: There is reduction in iterative shifting of dirty pages due to the use of historical based statistics of modification of pages for the forecast of frequent modified pages (WWS).

The same page repetitive shifting through iterative phase will be reduced if there is correct prediction of frequently updated pages. Because of this, during migration similar pages transfer many times has been reduced. For counting the page updation i.e. how many numbers of times a page has been modified TO_SEND_h [i] array had been used. A space of two bytes is being used for storage of each entry into TO_SEND_h [i] array. With the use of TO_SEND_h [i] array division into two groups of modified pages i.e. into G1 and G2 as per condition $PG1 < T1 \leq PG2$ is done.

Updated pages having value of updation less than a threshold value T1 are being considered less frequent updated pages and will be in G1, but updated pages with value of updation greater than or equal to T1 are being considered as frequent modified pages and will be in G2.Pages of group G1 after every iteration are shifted. This method provides dynamic value of threshold; it will be changed according to the following given formula at each of iteration:

$T1 = \lfloor[(\max[\text{page modification rate}] + \min[\text{page modification rate}]) \div 2]\rfloor$

In every iteration T1 is calculated with the help of using TO_SEND_h[i] array. Keeping pages which are frequently updated pages till the last iteration will reduce needless shifting of pages repetitively. Also it will cause reduction in total no of pages shifted and migration time.

Third Phase: This phase is stop and copy phase. During this phase, left over pages will be shifted as per two conditions. The first of the stopping condition is the same as is the stopping condition of precopy method. The second of the stopping condition states if the number of dirty pages in last iteration is greater than 1.5 times of the number of dirty pages in the previous iteration, the pages will be compressed and then transferred. Compression method will be Run Length encoding. This condition will be used for ensuring that downtime should be bearable.[8]

CHAPTER-3

PROBLEM FORMULATION

3.1 SCOPE

The performance of live migration techniques can be measured by following parameters:-

1. Total migration time: Total time taken place between migration initiation and the time the migrated VM start its execution at destination machine.

2. Downtime. It is time required to suspend VM running at source, shifting of VM to destination, starting migrated VM on remote host.

3. Total pages transferred: The total data is transferred while synchronizing the both VM's state

There are many studies proposed to minimize all the above parameters to improve the performance of live migration of VM.

The scope of proposed work is the optimization of precopy approach used for transferring the VM state considering combination of threshold and history of page modifications during 2nd phase of precopy method.During each iteration, first threshold value is calculated by taking the average of all the page modifications and then history of previous 3 iterations is taken into account. If a particular page has not been modified during last 3 iterations, that page will be migrated to target. The study will be repeated by taking the history of previous 2,3 and 4 iterations and results will be compared to find out the best. Also threshold calculation will be considered in two ways i.e. first by excluding the page that has not been modified as yet and second by including the page that has not been modified as yet. The results of both ways of threshold calculation will be compared to find the best.

3.2 OBJECTIVES OF THE STUDY:

1. To improve the performance of precopy live VM migration using two phase strategy.

2. To decrease total migration time of precopy live VM migration

3. To decrease downtime of precopy live VM migration

CHAPTER-4

METHODOLOGY

In order to implement the system, two systems are taken (Ref diagram). One will be called source(i.e. Host A) and other will be called destination(i.e. Host B).The source will have one virtual machine vm1 that will required to be shifted to destination (i.e. Host B) .

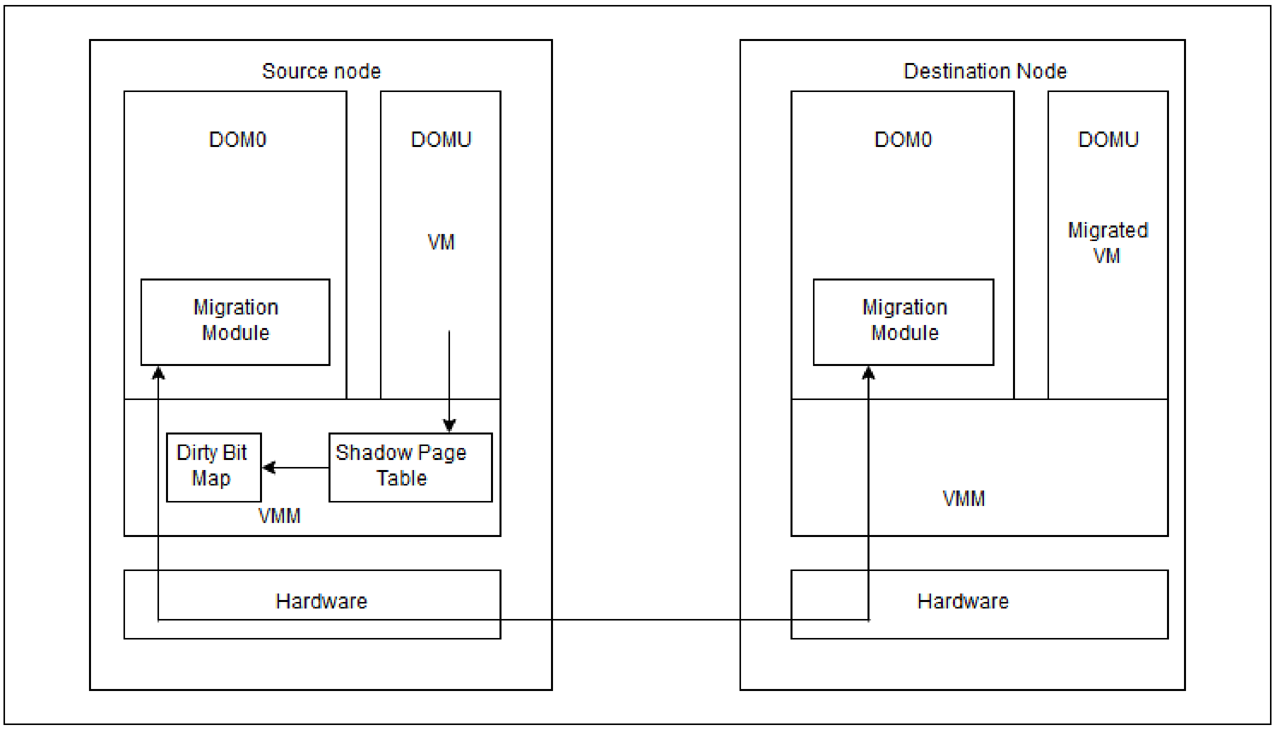


Fig: 4.1 Migration of VM from source to Destination

METHOD DESIGN:

First Phase: It is concerned with the shifting of pages of memory during first iteration of migration

Second Phase: (1) 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.

(2) History of last three iterations will be checked for the pages passed from the first stage.

Third Phase: In this phase comes last iteration.

. The three phases discussion given below:

First Phase: First phase of migration is dependent on three conditions:

1 Similar to precopy method all memory pages will be shifted if none of pages of memory has been modified. (That is virtually an extraordinary case).

2. Only pages that pass the test of historical statistics described in 2nd phase will be transferred if all of pages of memory has been modified. (That is also virtually an extraordinary case).

3 Only unmodified pages of memory will be transferred when there is fraction of pages of memory modification.

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

Second Phase: One array will be maintained which stores number of times each page has been modified/updated since the vm has started execution. During the second phase of precopy 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

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 pages 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.

The flowchart of proposed approach is given below:

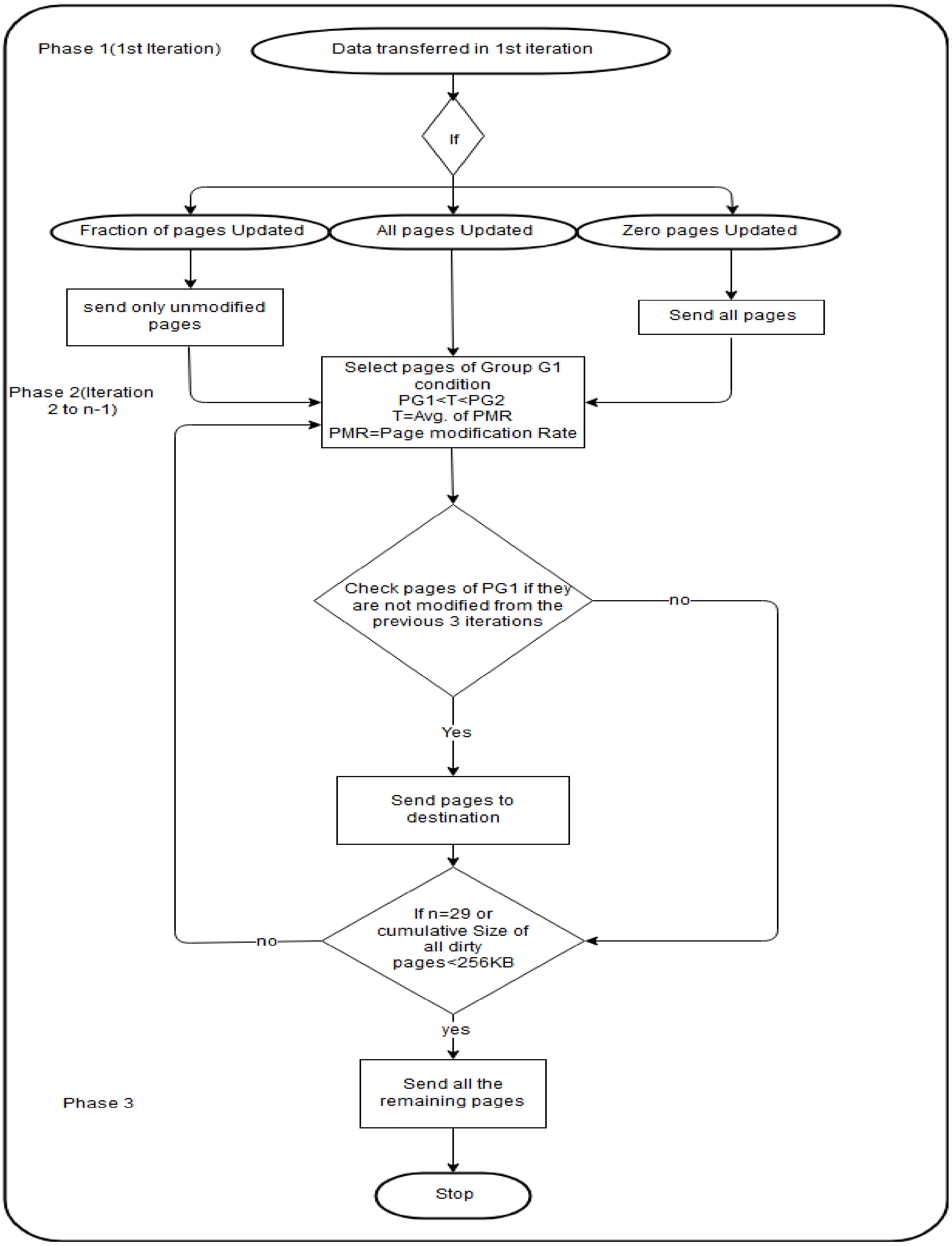

Phase 1(1st Iteration)
Data transferred in 1st iteration
If
Fraction of pages Updated
All pages Updated
Zero pages Updated
send only unmodified pages
Send all pages
Phase 2(Iteration 2 to n-1)
Select pages of Group G1 condition PG1<T<PG2 T=Avg. of PMR PMR=Page modification Rate
Check pages of PG1 if they are not modified from the previous 3 iterations
no
Yes
Send pages to destination
If n=29 or cumulative Size of all dirty pages<256KB
no
yes
Phase 3
Send all the remaining pages
Stop

Fig 4.2 Flowchart of proposed approach

CHAPTER-5

EXPECTED OUTCOME

The majority of hypervisors use precopy 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 (i.e. stop and copy) phase all the remaining pages will be transferred to destination machine. The problem with this method is the repeated transfer of dirty pages (same pages are transferred multiple times) during the second phase. This increases the total migration time and downtime. The proposed approach first calculates the threshold from the average of number of modifications of all pages. Only pages having modifications greater 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 whether to transfer the page or not. This strategy expects to decrease the number of dirty pages transferred to destination during iterative part of migration. So it will decrease the total migration time and downtime.

CHAPTER-6

SUMMARY AND CONCLUSIONS

As precopy is the frequently used method for live VM migration, it is very important that the total migration time, downtime and total no of pages transferred during the migration process should be minimum. As the cloud service provider has SLA with the customer and the provider has to pay the penalties in case of service disruptions that may occur due to high downtime and total migration time of VM migration. So its very important to make the precopy migration process as optimized as could be to give best performance. The proposed approach tries to achieve this objective.

LIST OF REFERENCES / BIBLIOGRAPHY:

1. 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

2. 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

3. 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)

4. 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

5. Danwei Chen,_, Hanbing Yang1, Qinghan Xue, and Yong Zhou" Live Migration of Virtual Machines Based on DPDT" China conference on wireless sensor networks CWSN 2012 pp26-33

6. 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

7. Parvin ahmadi doval amiri, Shayan zamani rad, Faramarz safi isfahani" Providing a solution to improve pre-copy Method for migrating virtual machines in Cloud infrastructure" Journal of Theoretical and Applied Information Technology 31st October 2016. Vol.92. No.2 ISSN: 1992-8645

8. Sangeeta Sharma and Meenu Chawla "A three phase optimization method for precopy based VM live migration" SpringerPlus (2016) 5:1022

9. 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

10. 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

11.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

12 Zhang X, Huo Z, Ma J, Meng D " Exploiting data deduplication to accelerate live virtual machine migration." In: 2010 IEEE international conference on cluster computing (CLUSTER). IEEE, pp 88–96

13 Diego Perez-Botero "A Brief Tutorial on Live Virtual Machine Migration from a security perspective"

14 Michael Nelson, Beng-Hong Lim, and Greg Hutchins" Fast Transparent Migration for Virtual Machines"ATEC '05 Proceedings of the annual conference on USENIX Annual Technical conference pages 25-26

15 Clark C, Fraser K, Hand S, Hansen JG, Jul E, Limpach C, Pratt I, Warfield A (2005) "Live migration of virtual machines. In: Proceedings of the 2nd conference on symposium on networked systems design & implementation-volume 2. USENIX Association, pp 273–286

16 Jin, Deng, Wu, Shi, & Pan"Live virtual machine migration with adaptive, memory compression" Published in: cluster computing and workshops 2009. Cluster '09 IEEE international conference.

17 Voorsluys, Broberg, Venugopal,and Buyya "Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation" Published in Proceeding CloudCom '09 Proceedings of the 1st International Conference on Cloud Computing Pages 254 - 265

18. Jon Oberheide, Evan Cooke, Farnam Jahanian"Empirical Exploitation of Live Virtual Machine Migration" Proc. of BlackHat DC convention 2008.

19 Raouf Boutaba Mohamed Faten Zhani "Virtual machine migration in cloud computing environments: Benefits, challenges, and approaches" https: //www.researchgate.net/ publication/ 283599580