A Survey on Reducing Time Of Virtual Machine Migration In Cloud

Nirali Y Raval
Silver Oak College of Engineering and Technology,
Gujarat Technological University

***Abstract*— Cloud Computing is one of the leading technologies. As a solution to many of the challenges faced by Cloud providers, virtualization is employed in Cloud. Virtual machine migration is a tool to utilize virtualization well. In general, the term "virtualization" refers to the process of turning a hardware-bound entity into a software-based component. The end result of such procedure encapsulates an entity's logic and is given the name of Virtual Machine (VM). The main advantage of this technique is that multiple VMs can run on top of a single physical host, which can make resource utilization much more e_cient. Of particular interest are those VMs with high availability requirements, such as the ones deployed by cloud providers, given that they generate the need to minimize the downtime associated with routine operations. VMs can deal with availability constraints much more gracefully than their physical equivalents. While physical hosts have to be powered down for maintenance, the VMs that they serve can migrate to execute on other physical nodes. It is also common to migrate VMs when load balancing is needed in the physical plane. The process of migrating VMs without any perceptible downtime is known as Live Virtual Machine Migration and is the topic of this paper. This nontrivial problem has been studied extensively and popular hypervisors (e.g. Xen, VMware, OpenVZ) have now put reasonable solutions to practice. After covering the pre-copy and post-copy approaches to live VM migration, a variety of design decisions will be discussed along with their pros and cons. Having set the necessary theoretical background, a security-focused survey will be carried out, documenting the state-of-the-art in Live VM Migration exploits and countermeasures.**

I. INTRODUCTION

Virtualization, a technique to run several operating systems simultaneously on one physical server, has become a core concept in modern data centers , mainly driven by benefit of application isolation, resource sharing, fault tolerance, portability and cost efficiency. In order to handle the requirements of various users, Cloud employs virtualization. In virtualization, there are entities called virtual machines which partition the available physical resources among various users. Virtualization provides many benefits. It allows efficient management of load, higher level of fault tolerance, energy efficiency and several others. Most of the benefits of virtualization are reaped by moving virtual machines across various physical hosts. This movement of a virtual machine (VM) from one physical machine to another is called Virtual Machine Migration. This is one of the widely researched areas. In some scenarios, the process of virtual machine migration is divided into 2 subtasks: virtual machine allocation and selection policy [2]. The allocation policy decides the physical host to which each virtual machine is assigned. The selection policy is responsible for deciding the target virtual machine, destination host of the migrating virtual machine and also the time at which migration should be performed. VM provides special benefit to server virtualization and has become a powerful tool for a variety of scenarios. Some of these include:

Power management: The aim is to consolidate virtual machines through live migration on an optimal number of servers and selectively switch off underutilized servers to reduce data centers power consumption.

IT maintenance: Administrators can transparently move virtual machines to free and shut down hosts for maintenance purpose.

Load balancing: The aim is to adjust virtual machine placement to achieve critical business goals, such as high throughput and high avalibity.

Although live migration is widely used, it does not come along without any negative impact, causes

performance loss of processes running inside VM as well as energy overheads .

II. BACKGROUND THEORY

Cloud Computing is one of the promising technologies that has attracted large attention over the past decade .Cloud Provide services and deployment model The deployment model decides how the services of the Cloud are provided to the users. There are 4 different deployment models: Private Cloud, Public Cloud, Community Cloud and Hybrid Cloud. Private Cloud is generally meant for a small group of users, say to a particular institution. On the other hand, public Clouds are generally open to the public. Community Cloud is shared by institutions or establishments that have some common interest. Hybrid Cloud is a combination of two or more of the aforementioned models. The service models give the different services provided by Clouds. Infrastructure-as-a-Service (IaaS) provides infrastructure such as physical machines, network and so on. Platform-as-a-Service (PaaS) provides compilers, databases and other such platforms required to run the applications. In the Software-asa- Service (SaaS) model, software applications are provided to users. Basically, IT is provided as a service under Anythings- a-Service (XaaS). This can also be a combination of any of the aforementioned service models.

Virtualization provides facility to migrate virtual machine from one host (source) to another physical host (destination). Virtual Machine Migration (VMM) is a useful tool for administrator of data center and clusters, It allows clean separation between hardware and software. Process level migration problems can be avoided by migrating a virtual machine. Virtual Machine Migration enables energy saving, load balancing and efficient resources utilization. Virtual Machine Migration methods are divided into two types: Hot (live) migration and cold (non-live) migration. Virtual machine keeps running while migrating and does not lose its status. User does not feel any interruption in service in hot (live) migration. The status of the VM loses and user can notice the service interruption in cold migration In live migration process, the state of a virtual machine to migrate is transferred. The state consists of its memory contents and local file system. Local file system need not be transferred. First, VM is suspended, then its state is transferred, and lastly, VM is resumed at destination host.

PERFORMANCE METRICS

The performance of any live VM migration strategy could be gauged by the following metrics .[5]

(1)Preparation Time: This is the time between initiating migration process and transferring the VMs processor state to the target node, during which the VM continues to execute and dirty its memory.

(2) Down Time: This is time during VMs execution is stopped .It includes the transfer of processor state..

(3) Pages Transferred: This is the total amount of emory pages transferred, including duplicates, across all of the above time periods.

(4)Resume Time: This is the time between resuming the VMs execution at the target and the end of migration, all dependencies on the source are eliminated.

(5)Total Migration Time: Total time taken by migration process from start migration process to finish the migration process. Total time is very important because of it affects the release of resources on both source and destination nodes.

(6)Application Degradation: When Virtual machine migrated from one host to another, the application performance is degraded which is running on that vm.

There are two major approaches: Post-Copy and Pre-Copy memory migration[1]. In the Post-copy approach first suspends the migrating Virtual Machine at the source side then after copies minimal processor state to the target host and resumes the virtual machine, and begins fetching memory pages over the network from the source node.

There are two phases in Pre-copy approach: Warm-up phase and Stop-and-Copy phase. In warm up VM memory migration phase, 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 during memory copy process dirty pages, they will be re-copied until the rate of recopied pages is not less than page dirtying rate. In Stop and Copy phase, the VM will be stopped in source and the remaining dirty pages will be copied to the destination and VM will be resumed in destination.

Pre-Copy Phase: At this stage, the VM continues to run, while its memory is iteratively copied page wise

from the source to the target host. Iteratively means, the algorithm works in several rounds. It starts with transferring all active memory pages. As each round takes some time and in the mean time the VM is still running on the source host, some pages may be dirtied and have to be resent in an additional round to ensure memory consistency.

Pre-Copy Termination Phase: Without any stop condition, the iteratively pre-copy phase may carry on indefinetly. Stop conditions depend highly on the design of the used hypervisor, but typically take one of the following thresholds into account: the number of performed iterations exceeds a pre-defined threshold , the total amount of memory that has already been transmitted, exceeds a pre-defined threshold.

Stop-and-Copy Phase

At this stage the hypervisor suspends the VM to stop page dirtying and copies the remaining dirty pages as well as the state of the CPU registers to the destination host. After the migration process is completed, the hypervisor on the target host resumes the VM.

III. LITERATURE REVIEW

Live migration enables increased flexibility in provisioning of resources and current VM hypervisors have support for live migration with downtimes as low as tens of a second when migrating over Local Area Networks (LANs).Wide Area Network (WAN) migration, as demonstrated by Ramakrishnan et al. [11] and Travostion et al. [14] usually involves both longer migration times and downtimes. In order to live migrate a VM its runtime state must be transferred from the source to the destination with the VM still running. If the VM's file system is kept on a network share accessible to both source and destination it need not be migrated. This is difficult in cross-site and WAN migration so in these cases the file system needs to be migrated. In this contribution we only consider memory migration, but our algorithms could be adapted for storage migration. A Typical live migration algorithm There are several variations of the live migration algorithm but most implementations share the same basic idea, first hypervisor marking all memory pages as dirty. The algorithm then iteratively transfers dirty pages over the network until the number of pages remaining to be transferred is below a certain threshold or a maximum number of iterations is reached. Transferred pages are marked as clean by the hypervisor. Notably, as the VM operates as usual during live migration, already transferred memory pages may be dirtied during an iteration and must thus need to be re-transferred. To stop further memory writes and enable transfer of the remaining pages, the VM is at some point suspended on the source. When the complete memory contents has been transferred, the VM is resumed at the destination and the live migration is complete. There are two important performance criteria for (live) migration: migration downtime and total migration time. In addition to transparency to users in terms of non interrupted service operation, other requirements for live migration include low impact on the performance of the running VM and any co-hosted VMs. If the live migration process uses too much system resources, VM performance suffers and in the worst case, service is interrupted. Another requirement is that live migration should be transparent to VMs and applications running inside the VMs so that they need not be migration aware in any way.

To evaluate the performance of our algorithms, we perform a series of live migration tests with VMs running two kinds of workloads with varying working set sizes. In these tests, the standard KVM Algorithm, denoted *Vanilla*, is compared to the algorithm proposed in this paper, denoted *PRIO*, where dynamic page transfer reordering is combined with delta compression, and a version with only the delta compression modification, henceforth called *XBRLE* (XOR Binary Run- Length Encoding). *A. Experimental scenarios*

To put load on the VMs, the LMBench [2] benchmarking software is used. The LMBench benchmark generates a very high page dirtying rate by allocating a big block of memory and then continuously overwriting the memory contents through a series of 4 byte store and increments. Using several instances of LMBench as workload, we vary the size of the working set and perform a series of migrations where we measure migration downtime, total migration time, amount of data transferred, and number of page resends. Finally, to evaluate the PRIO algorithm's performance in

a real world scenario, we live migrate a streaming video server over a limited bandwidth link to simulate a crosssite migration. The streaming video scenario is an example of a real world application where the memory data already is compressed as the video buffer is in h.264 format in our case. The details of the setup of the experimental scenarios are presented in Table I.

TABLE I
SUMMARY OF EXPERIMENTAL SCENARIOS.

Scenario	VMSize	Workload	Network	Algorithms
Migration downtime	2GB, 1vcpu	LMBench	1000 Mbits/s	Vanilla, XBRLE PRIO
Total migra-tiontime	2GB, 1vcpu	LMBench	1000 Mbits/s	XBRLE, PRIO
Streaming video	512MB, 1vcpu	VLC video server	100 Mbits/s	Vanilla, PRIO

IV. CONCLUSION

Current approach for live migration, the pre-copy paradigm are explained.There are many techniques which attempt to minimize the down time and provide better performance in low bandwidth environment.We have categorized the papers and there is a need to compare techniques in each category to understand the strengths and weaknesses. In future, we plan to propose a performance model based on the research gaps identified through the limitations. This will be helpful for reducing the migration time with heavy workload.

REFERENCES

[1] Shoaib Hassan, Asim Abbas Kamboh, Dr. Farooque Azam. Analysis Of Cloud Computing Performance, Scalability, Availability, & Security 978-1-4799-4441-5/14/$31.00 ©2014 IEEE

[2] Migration, Christina Terese Joseph, Prof. K Chandrasekaran, Robin Cyriac. A Perspective Study Of Virtual Machine 978-1-4799-3080-7/14/$31.00_c 2014 IEEE

[3] J.Arputharaj Johnson. Optimization Of Migration Downtime Of Virtual Machines In Cloud th ICCCNT 2013 July 4-6, 2013, Tiruchengode, India IEEE

[4] Petter Sv¨ard and Johan Tordsson,Benoit Hudzia,Erik Elmroth High Performance Live Migration Through Dynamic Page Transfer Reordering And Compression 2011 Third IEEE International Conference on Coud Computing Technology and Science.

[5] Patrick Raad, Stefano Secci, Dung Chi Phung, Antonio Cianfrani, Pascal Gallard, and Guy Pujolle. Achieving Sub-Second Downtimes in Large-Scale Virtual Machine Migrations with LISP 1932-4537/14/$31.00 _c 2014 IEEE

[6] Adel Amani, Kamran Zamanifar. Improving the Time of Live Migration Virtual Machine by Optimized Algorithm Scheduler Credit.

[7] P. Kokkinos, T.A., A. Kretsis, E.A. Multi-Criteria Virtual Machines Migration Considering the Reconfiguration of their Logical Topology. 2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems

[8] Rabiatul Addawiyah Mat Razali, Ruhani Ab Rahman, Norliza Zaini, Mustaffa Samad. Virtual Machine Migration Implementation In Load Balancing For Cloud Computing, 978-1-4799-4653-2/14/$31.00 © 2014 IEEE.

[9] Mahdi Aiash, Glenford Mapp, Orhan Gemikonakli. Secure Live Virtual Machines Migration: Issues and Solutions. 2014 28th International Conference on Advanced Information Networking and Applications Workshops

[10] Uttam Mandal, M. Farhan Habib, Shuqiang Zhang, Pulak Chowdhury, Massimo Tornatore†, and Biswanath Mukherjee Heterogeneous Bandwidth Provisioning for VirtualvMachine Migration over SDN-Enabled Optical Networks

[11]Chungmu Oh and Won Woo Ro. Hyper Threading-aware Virtual Machine Migration