www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 4 April 2017, Page No. 20806-20808
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i4.01

Optimization of Migration Time affected storage overheads during VM Live Migration using Network Attached Storage device

Preeti P. Thakre[1], Vaishali N. Sahare[2]
[1] Department of Computer Science and Engineering, G. H. Raisoni Institute of Engineering and Technology, RTMNU University, Nagpur, Maharashtra
preetithakre28@gmail.com
[2] Department of Computer Science and Engineering, G. H. Raisoni Institute of Engineering and Technology, RTMNU University, Nagpur, Maharashtra
vaishali.sahare@raisoni.net

Abstract: ***Cloud computing has grown in relevant amount from recent years by inheriting a key feature of server virtualization which is the live migration of virtual machines(VMs). This technique mainly used in datacenters in order to provide cloud services without letting servers down by allowing a virtual machine from one system to another with minimum service interruption. Due to limited available network bandwidth, the network traffic get increases due to large amount of storage data which lead to create overheads during migration and hence increases migration time. This paper focuses on such type of duplicated data which consumes only unnecessary shared network bandwidth, increases migration time during live VM Migration and provide optimized solution using proposed scheme with the help of Network Attached Storage Device (NAS).***

Keywords: **Cloud Computing, Virtual Machine, Live Migration.**

I. Introduction

In cloud computing, the technology that are acceptable to provide virtualized view of physical resources (like for starting a VM) and providing services such as live migration by accommodating a number of virtual machines which refers to term known as Virtualization. Virtualization has become an integral component of the modern data center. It uses software to simulate the existence of hardware and create a virtual computer system. Doing this allows organization to run more than one virtual systems and applications on a single server. This can provide economies of scale and greater efficiency[6]. Live migration is a technology used by administrators to take a host offline for maintainence or upgrading without it. Live migration is mainly used in datacenters for various purposes such as freeing up resources for save power, resource sharing and system maintainence. In general, live migration of running virtual machines (VMs) across distinct physical machines (PMs) serves as the milestone for load balancing and power saving in modern datacenters. Live migration of VMs is most promising approach to control performance overhead of PMs.

A. VM Live Migration and it's process:

Migrating the virtual machine from one physical server to another either within the same computing environment or across different datacenters is known as Live Migration. During VM live migration, a lot of data like CPU state, memory state, storage state and network state of the targeted virtual machine has been migrated via a computer network (LAN or WAN) from one system to another system.

B. Phases of Live migration

It has two phases a) Pre-copy Migration Phase and b) post-copy Migration Phase as shown in Fig.1.

Efficient VMs live migration can achieve effective VM consolidation which is unique feature to reduce the energy consumption of cloud data centers by putting some of PMs in sleeping mode (power saving mode), while focusing on QoS (Quality of Service) objective. In pre-migration, all the memory pages copies from source to the destination. During this process, there can be modification in the memory pages by user or operating system called as dirty pages. These dirty pages transfers until it will not equal to or exceed the modification rate. Once it reached, migration manager will suspend the VM on source, transfer the remaining dirty pages and alternatively, resume the VM on destination machine.

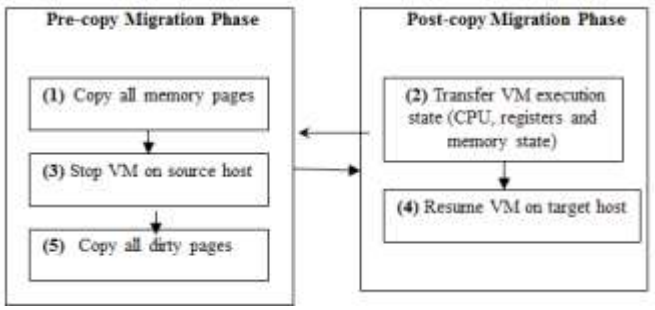

Fig.1. Phases of Live Migration

Live migration of virtual machine is usually done by suspending VM at the source and resume at the destination after transferring the copied VM states of the VM at the source machine. The time period during which the source VM or domain stops or suspended is known as downtime and the time taken by the VM to migrate along with its states from source to the destination is known as migration time. It takes several iterations about 20-29 iterations and in each last iteration about 40-50 pages dirtied and almost 3 times RAM has been copied which fortunately ranges downtime from milliseconds to seconds on source machine which is comparitively shorter than migration time and downtime of destination machine.

II. Literature Review

In 2015, U. Deshpande, K. Keahey [1] addresses the problem of network contention during live migration of co-located virtual machines and proposed approach of using the combination of pre and post-copy techniques at source host and destination host instead of any single predetermined technique.

In 2015, D. Li, J. Zhu, J. Guan, Y. Zhang [2] proposed a heterogeneous bandwidth request model to control bandwidth demand and VM placement. It provides more flexibility for tenants to abstract their needs and VM allocations algorithms to solve this NP-complete problem of network overheads created by VM migration.

In 2015, J. Hwang, K. K. Ramkrishnan and T. Wood [3] proposed NetVM which is a high-speed network packet processing platform where Simple descriptor need copied via shared memory gives VMs direct access to huge size packets. This approach improves throughputs for complex network functionality.

In 2016, J. Guo, J. Lui, H. Jin [4] used a bandwidth allocation algorithm Falloc in order to achieve the asymmetric Nash Bargaining Solution (NBS) in datacenter networks to achieve bandwidth guarantee and fairness in sharing. It provides flexible fairness for Vms by balancing the tradeoff between bandwidth guarantee and proportional bandwidth share.

In 2013, H. Guan, Y. Dong, R. Ma, Y. Zhang [7] implemented an interrupt coalescing and virtual RSS to enhance performance for network virtualization. It is able to obtain a 2.6" bandwidth improvement over the baseline in the high-speed (10Gbps) network environment.

In 2013, Z. Wu, Y. Zhang, G. Jiang [8] implemented NetDeo, a swarm intelligence based optimization model for topology independent resource allocation and optimization. It achieved efficient VM placement in a data center, optimal deployment plans for newly added Vms, Hardware upgrade suggestions and allowing the computer network to evolve as the workload changes over time.

III. Challenges during Live Migration

Since the access latency of persistent storage systems is still several orders of magnitude slower than that of volatile memory chips, modern operating systems aggressively cache data from the storage system in memory in order to hide the long access latency. Therefore, there is a large portion of data cached in memory with a duplicated copy in the storage system.

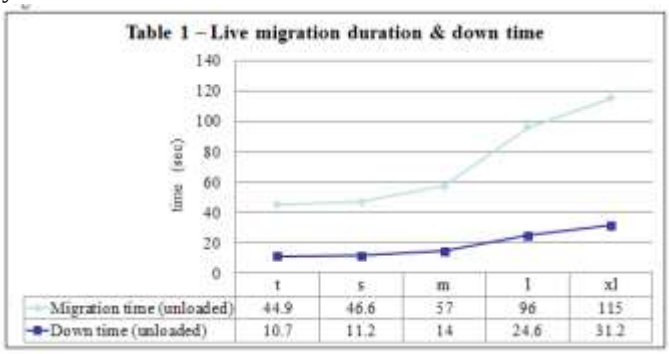

Fig.2. Analysis of Migration Time and Downtime during VM Live Migration

There is too much amount of data duplication between memory and storage device about 50-60% which takes too much time to transfer during migration as compared to downtime as shown in Fig2. Therefore, it is necessary to track the VM's I/O operations by using network-attached storage device and maintains an updated mapping of memory pages that currently reside in identical form on the storage device.

IV. Proposed solution to optimize storage overheads using Network Attached Storage (NAS) device

As it also increases migration time, therefore during migration instead of transferring or migrating whole data from the memory, the key idea is to directly fetch such data from the Network attached storage device by destination machine after migration as shown in Fig.3.

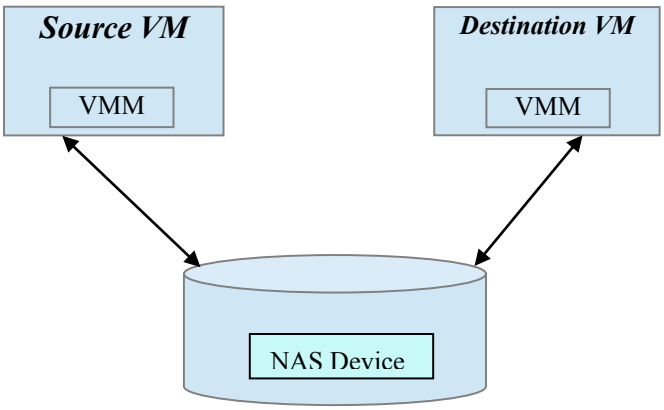

Fig.3. Fetching duplicated data directly from Network attached storage device (NAS)

As a result, data transmission rate and migration time will reduced. So, there is need to first track and analyze the migration time and downtime before creating NAS on the QEMU/KVM platform. As shown in Fig.4., the tracking of the VM's I/O operations using NAS is done by keeping duplicated data. After accepting the request for live migration from server, there is need to first analyze the how much amount of data is going to transfer and second it is necessary to send complete data including duplicated as well as unique by Central Cloud Controller as shown in Fig.5.

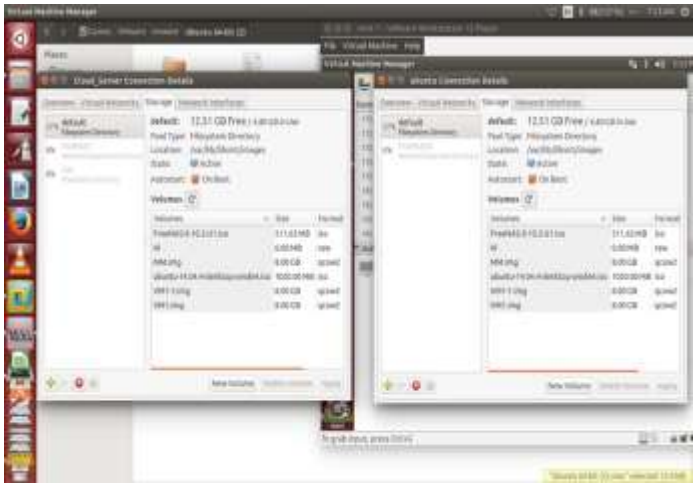

Fig.4. Duplicated data in NAS after mounting with source and destination hosts

In such case, there is need to propose a scheme which will maintain an updated memory pages of RAM and NAS during migration and then compare each and every page. If during comparison data in the both storages found same, then central cloud controller will keep that duplicated data in NAS as shown in Fig.4. and will signals to Migration manager to migrate only unique data. Therefore, during the pre-copy migration process instead of transferring duplicated pages from the source to the target host, only memory-to-disk mapping is sent to the target host which then fetches the contents directly from the network-attached storage device. Due to the large computational overhead, the detection of duplication between memory pages and disk blocks immediately before transmission is difficult. Therefore, there is need to keep a hash value to every disk block and memory pages by *Central cloud controller*. There is need to re-map the selected pages in order to detect modification in memory pages of RAM as well as disk blocks.

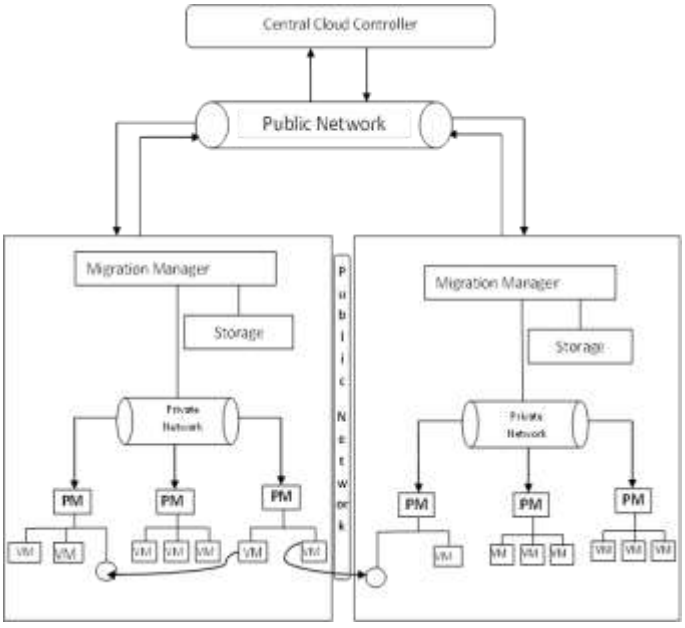

Fig.5. Proposed System Architecture

Since the first phase of live migration is pre-copy, therefore the contents of the running VM's memory are sent to the target host over several iterations in iterative pre-copying approach. The proposed technique which is iterative pre-copy approach instead of sending all dirtied memory pages to the destination host, only pages whose contents are unique on external storage are transferred as shown in Fig.6. Dirty pages are those pages in which data modifies and stored in buffer cache. Such pages has not been written into the hard disk. The list of duplicated pages is assembled by joining the bit vector containing the list of dirtied pages with the current mapping of memory pages to disk blocks.

V. CONCLUSION

Since Live Migration is a very effective approach in datacenters for resource utilization load balance, server maintainence but it takes a lot migration time due to overheads. This paper focused on one of such overheads which is storage overhead, it takes migration time proportional to amount of storage data. It also find out solution to optimize such overheads which are responsible for
increasing migration time with the help of proposed scheme using Network Attached Storage device by implementing it on the QEMU/KVM platform.

REFERENCES

[1] U. Deshpande, K. Keahey, "Traffic-sensitive live migration of virtual machines," IEEE/ACM International Symposium on May 2015.
[2] D. Li, J. Zhu, J. Guan, Y. Zhang, "Guaranteeing heterogeneous bandwidth demand in multitenant data center networks," IEEE/ACM Transactions on Networking(TON) 23.5 (2015): 1648-1660.
[3] J. Hwang, K. K. Ramkrishnan and T. Wood, "NetVM:high performance flexible networking using virtualization on commodity platforms," IEEE Transactions on Network and Service management 12.1 (2015): 34-47.
[4] J. Guo, J. Lui, H. Jin,"Fair network bandwidth allocation in Iaas datacenters via a cooperative game approach," IEEE/ACM Transactions on Networking 24.2 (2016): 873-886.
[5] U. Deshpande, Y. You, D. Chan, N. Bila, K. Gopalan, "Fast server deprovisioning through scatter gather live migration of virtual machines," IEEE International conference on cloud computing, 2014.
[6] W. Zhang, K. Lam, C. Wang, "Adaptive live migration over a WAN: Modelling and Implementation," IEEE International conference on cloud computing 2014.
[7] H. Guan, Y. Dong, R. Ma, Y. Zhang et al," Performance enhancement for network i/o virtualization with efficient interrupt coalescing and virtual receive-side scaling," IEEE Transactions on Parallel and Distributed Systems 24.6 (2013): 1118-1128.
[8] Z. Wu, Y. Zhang, G. Jiang, "Automating cloud network optimization and evolution," IEEE Journal on Selected Areas in Communications 31.12 (2013): 2620-2631.