Research Journal of Recent Sciences _____ ISSN 2277-2502
Vol. 3**(IVC-2014),** 13-20 **(2014)** *Res. J. Recent. Sci.*

An Efficient Live VM Migration Technique in Clustered Datacenters

Kumar Narander and Saxena Swati
Department of Computer Science, B. B. A. University (A Central University), Lucknow, UP, INDIA

Available online at: www.isca.in, www.isca.me
Received 4th March 2014, revised 21st August 2014, accepted 9th September 2014

Abstract

The essence of cloud computing comes from the concept of virtualization. It is a technique of implementing a number of guest operating systems on a single host server such that memory and CPU resources of the host machine are shared among the guest systems. The guest machines are technically known as virtual machines (VMs). Virtualization enables full utilization of a physical machine in a cost-effective manner. However, varying and continuous load from users may overload a physical machine and this can lead to serious implications on performance, reliability and other service-level-agreement (SLA) parameters. To deal with this issue, VM migrations are practiced which enables the transfer of a virtual machine from an overloaded server to an under-loaded host, thereby relaxing the workload of the source host. Load balancing in a datacenter gives a chance to achieve a significantly high fault-tolerance, a feature which is very essential during live application executions in a cloud environment. This paper discusses the basic live migration techniques existing today, and proposes a hybrid approach of live VM migration, by combining the benefits of other existing techniques. Live migration refers to a transparent transfer of an active guest or virtual machine from a source server to a chosen destination server.

Keywords: Cloud computing, datacenters, virtual machine migration, load-balancing.

Introduction

The popularity of cloud computing revolves around the concept of virtualization. This technique enables multiple operating system instances to run concurrently on a single physical machine, thereby, separating hardware from a single operating system. Each guest 'operating system' is managed by a virtual machine monitor (VMM), also known as the hypervisor. To control a guest operating system, Virtualization allows an operator to use memory, CPU, storage and other resources, so that these resources are managed efficiently among the guest operating systems. Since the guest operating system is not bound to the hardware, it is possible to dynamically move an operating system one physical machine to another. This concept is known as virtual machine (VM) migration. VM migration can be considered as the lifeline of a cloud datacenter. As a particular guest OS or VM begins to consume more resources during a peak period, it can be moved or migrated to another physical machine with less demand[1]. Migration of a VM seeks to improve manageability, performance and fault-tolerance of a system. More specifically, it justifies the load-balancing feature of a datacenter by migrating VMs out of an over-loaded server to an under-loaded server or physical machine. It can also help in energy saving of a datacenter by allowing under-utilized servers to be shut down by migrating their VMs to other optimally utilized servers. The key secret of a VM migration is that the application itself or its corresponding processes are unaware that the VM migration is taking place, thus, infusing the notion of transparency. Popular hypervisors, such as Xen and VMWare, allow migrating an OS as it continues to run[2]. Such procedure is known as 'live' or 'hot' migration, as opposed to 'pure stop-and-copy' or 'cold' migration, which involves halting a VM, copying all its memory pages to the destination machine and then restarting the new VM. The main advantage of 'live' migration is near-zero downtime, which is an important feature when live services are being served in a cloud datacenter[3].

This paper discusses the existing live migration techniques and proposes a hybrid migration methodology in clustered environment which not only reduces the migration time but also improves the service downtime considerably.

Related work: A number of approaches have been discussed to improve live VM migrations in a cloud datacenter. Their aim has been either to make the datacenter energy efficient or improve the fault tolerance of the cloud environment or to achieve reliability by balancing the uneven load efficiently.

Timothy Wood et al. introduce a system that monitors and detects VMs utilizations of resources on servers so as to initiate migrations whenever and wherever necessary. They also advocated the use of VM swapping along with migration so as to balance the uneven load on the cloud datacenter. Extending the concept of migrations, Anja Strunk and Waltenegus Dargie study the power consumed and the time taken by virtual machine migrations. The authors conclude that energy consumed during live VM migrations depends on the VM size and the available network bandwidth. These two factors effect the performance of migrations considerably[2].

On the similar lines, William Voorsluys et al. monitor the effect of live VM migrations on Xen machines[3]. They insist on the fact that live VM migrations come with certain overheads which are not negligible. The aim should be to minimize these overheads to its minimum. Diego Perez-Botero explains the procedure of existing pre-copy and post-copy techniques with their advantages and disadvantages[4]. Author has established that the post-copy approach comes with certain benefits over pre-copy, however, is not suitable for VM migration in WAN. Samer Al-Kiswany et al. present a new migration technique called VMFlockMS, which is meant for VM migrations across different data centers in groups. Authors have suggested an application-level solution, for e.g., a three-tier web application[5].

Stressing on the fact that VM migrations must be optimized for multiple factors, authors Wei Deng et al. propose RACE, a Reliability-Aware server Consolidation strategy, which decides when and how to perform energy-efficient server consolidation in a reliability-friendly and profitable way[6]. They have developed a system which combines various execution metrics such as reliability, energy- efficiency, availability, etc. in a holistic manner. In order to speed-up the migration process for both LAN and WAN environments, authors Hai Jin et al. implement a new VM migration system which uses check-pointing/recovery and trace/replay fundamentals. Thereafter, the source and the destination virtual machines are tuned to a consistent state[7].

A comparison of pre and post-copy live VM migration techniques is highlighted by authors Michael R. Hines and Kartik Gopalan, while applying them on Xen Hypervisor. Further they propose a dynamic self-ballooning system, DSB, which improves the migration speed by releasing back unused memory from a guest virtual machine to the hypervisor or VM manager[8]. This method does degrade the datacenter performance within the acceptable limits. In order to improve the Quality of Service (QoS), authors Dejene Boru et al. try to reduce the communication delays by replicating data in cloud computing datacenters so that the energy efficiency and bandwidth consumptions are also improved[9]. Alexander Stage and Thomas Setzer group VMs based on their workload and then for each group, authors suggest proper resource and migration scheduling models based on the group's topology and bandwidth requirements[10].

Erik Gustafsson modifies the migration technique by not sending duplicate data from the disk so that the new migrated VM can be resumed before full memory transfer has taken place at the destination host[11]. Meng Sun et al. introduces a model where number of physical machines, on which VMs need to be placed, is reduced and also the resource utilization of each physical machine is optimized[12]. Kamran Zamanifar et al. emphasize that the location of virtual machines and their data file sizes play an important role in proper utilization of cloud datacenter resources[13]. T. Hirofuchi et al. considers live VM migration in wide area networks and present two techniques to manage the input-output consistency of virtual disks before and after migration[14]. A. Khosravi et al. concentrate on increasing carbon footprints and power usage of cloud datacenters and propose a migration mechanism to reduce them effectively[15]. On the same lines, A. Verma et al. present pMapper, an architecture where VMs are placed on servers after considering their cost requirements and power usage[16]. A. Beloglazov et al. propose a migration technique of VMs in groups so as to switch off servers which are idle or under-utilized, thereby minimizing power consumption and at the same time providing required QoS[17]. A. Gandhi et al. employ ERP (Energy-Response time Product) metric for studying the power consumption and execution-related behaviors of static user demands and also dynamic user demands in a datacenter[18]. A dynamic grouping of servers and migration of VMs reducing the SLA violations is introduced by N. Bobroff et al.[19]. T. Ferreto et al. prioritize VMs based on their steady capacities and reduce unnecessary migrations due to varying workloads[20].

Almost all these papers mentioned above, advocate the need of a live VM migration approach that should reduce the service downtime so as to fulfill all factors of service-level agreements (SLAs). In our paper, we have proposed a hybrid technique that is implemented in a clustered data center environment. Further, VM migrations are limited within the cluster. This helps in reducing the migration time and the volume of data to be migrated, as a cluster contains VMs belonging to same application. This approach proves to be a huge improvement over the approaches mentioned above, without compromising with any performance issues.

Comparison between Pre and Post-Copy live VM Migration Approaches: There are three main physical resources that are considered during live migration- memory, network and disk. Memory migration is divided into three sequences, as given below: i. Push step, ii. Stop-and-copy step, iii. Pull step

Generally, VM live migration strategies either combine push with stop-and-copy phases, which is known as the pre-copy approach; or they combine pull with stop-and-copy phase, naming it as the post-copy approach[4]. Figure 1 shows the essential difference between the two. The pre-copy approach uses an iterative *push phase*, starting with a low network bandwidth use and sending relatively read-only memory pages without compromising with the quality of service. Afterwards, more bandwidth and CPU resources are allocated to migration process incrementally to transfer more frequently-updated memory pages. This is done to reduce the number of hot pages (frequently updated pages) to a minimum before the *stop-and-copy phase* begins, however it also compromises the maximum throughput of migration for a small period of time. In this stop-and-copy step, source VM is halted and remaining memory pages are transferred to the destination physical

machine in one go. After this, the migrated VM is restarted at the destination machine and is synchronized with the VM at the source machine. After the synchronization is over, the new VM processes the input-output requests in the usual manner and the old VM at the source machine is shut down. Contrary to the pre-copy approach, the post-copy approach first suspends the victim VM at the source machine, transfers a minimum required processor state (*Pull phase*) to the destination machine, and then pulls the remaining memory pages from the source over the network. The main benefit of this approach is that the memory page is transferred once at the most, thereby reducing the overhead of duplicate transmission which was evident in the pre-copy scenario.

As evident from the above explanation, a source virtual machine's memory is copied directly to destination machine during migration, however, network interface and local disk migrations are not that simple. There are few important points to consider, such as for network interface migrations, a VM should retain its original IP address after migration so as to preserve open network connections and to avoid network redirection mechanisms[4].

Also, local disk migration should not be needed inside a server cluster. Datacenters use network-attached-storage (NAS) devices, which can be accessed from anywhere inside a cluster. Thus, by using a clustered approach, local disk migration is not needed[4]. Figure 2 shows a typical cloud datacenter architecture which will be the point of reference in our proposed technique also.

Now, live VM migration problem is reduced to transferring VM memory state from source host machine to a chosen destination host machine in the best possible manner. This paper proposes a hybrid approach of live VM migration in a clustered environment, exhibiting advantages of both pre-copy and post-copy methods.

Proposed System Model: Cloud datacenters typically implement a fat-tree topological structure. For the purpose of generality, we have presented a datacenter architecture containing three layers of switches, as shown in figure 2. The top-most layer consists of core switches, followed by layer two of aggregate switches. The third layer consists of access switches, which are attached with clusters. Every cluster is a group of physical machines or servers on which virtual machines mount. VMs belonging to same or identical applications are mounted on servers in the same cluster. Apart from switches, physical machines, VMs and links, a cloud datacenter also possess a Network-attached Storage (NAS) device.

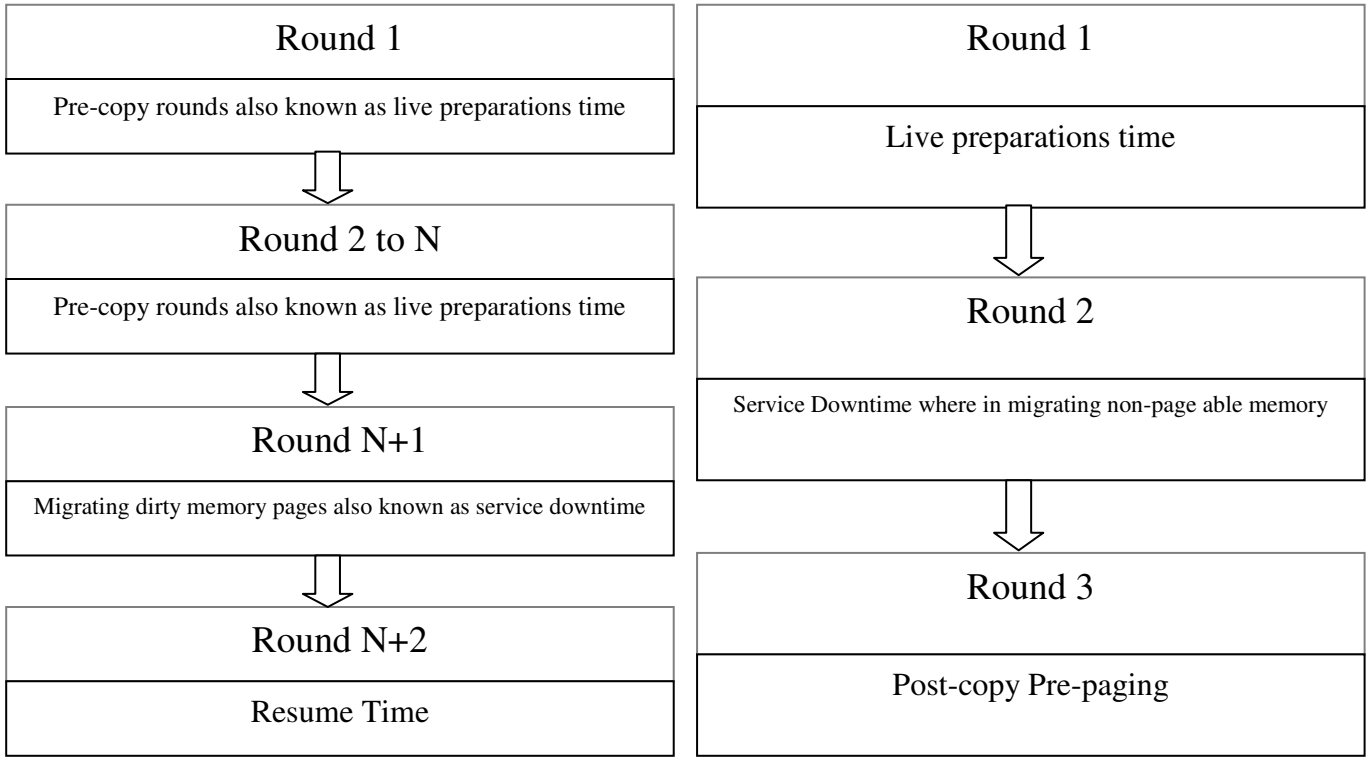

Figure-1
Timelines of pre-copy and post-copy live VM migration approaches

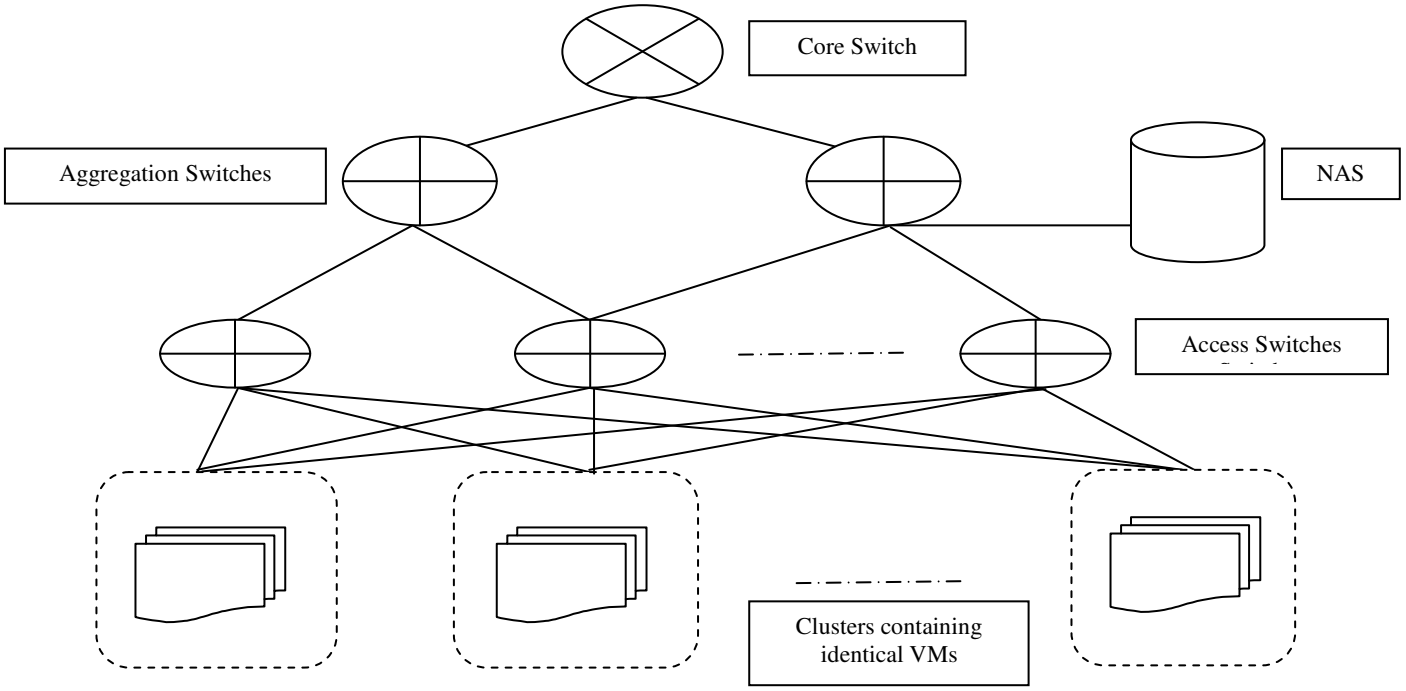

Figure-2
Typical Cloud Datacenter Architecture

The Process: Applications deployed on the cloud often involve multiple VMs with different images, for ex- a 3-tier web application, a business analytics solution, etc. The VM images belonging to the same application are often based on the same OS distribution, yet with different installed applications, leading to similarity ratio across images as high as 96%[5]. This will reduce the volume of data to be sent from source host to the destination host during live VM migration. Keeping this mind, we propose the following steps to ensure a fast and consistent live VM migration-

Clustered architecture: Exploiting the crucial feature of an application having identical VMs, we partition servers/hosts of a datacenter into clusters. Each cluster of hosts consists of VMs from the same or identical applications. Further, we are limiting the migration of any VM within its cluster only, so that the requirement of total image transfer is not needed as the destination machine will have a more or less identical VM image. Also, since NAS devices can be accessed from anywhere within the cluster, VM migration does not require transfer of local disk, as explained in the previous section. Moreover, there is no need for a new network address for the migrated VM as its original IP address will be preserved within the cluster, further reducing the overhead of network redirection mechanisms. Thus, we see that a clustered approach solve a number of migration issues which were existing in the previous works.

Monitoring and Selecting the server for migration: After arranging the datacenter in a clustered environment and placing VMs of one application in a particular cluster, we, now, monitor the load on every host in the cluster. This continuous monitoring will help in deciding whether or not to perform VM migration. For this purpose, we have selected four major resources of a server which are used by any VM

$$Load = 1/ (1-\Theta_{CPU})*(1-\Theta_{mem})*(1-\Theta_{disk})*(1-\Theta_{net})$$

here $\Theta_{CPU,}$ Θ_{mem}, Θ_{disk} and Θ_{net} are the corresponding average utilization ratios of memory, CPU, network bandwidth and disk for a particular host[6] and ranges between 0% and 1%. When this average load exceeds the maximum threshold for a particular duration T, our technique decides to migrate a VM from this server to another server in the same cluster.

For eg: In an idle server, where CPU, memory, disk and bandwidth usages are minimum, the corresponding average utilization ratios tend to become minimum. Hence,
Load = 1 which is minimum.

However, suppose the server is at 50% utilization by the VM, by all the above stated resources, then
$\Theta_{CPU} = 0.5$
$\Theta_{mem} = 0.5$
$\Theta_{disk} = 0.5$
$\Theta_{net} = 0.5$
In this case, Load = 1/ (0.5*0.5*0.5*0.5) = 16

Next, suppose the server is at peak resource utilization condition, i.e,
$\Theta_{CPU} = 1$
$\Theta_{mem} = 1$
$\Theta_{disk} = 1$

$\Theta_{net} = 1$
In that case, Load = 1/ 0

$= \infty$, which clearly shows that the server is overloaded. In this scenario, our method decides to migrate a suitable VM from this server to ease its load.

Selecting a suitable VM to migrate: Next, we establish the criteria for deciding which VM to migrate. A number of approaches are available[7]. We are considering the VMs of a cluster in descending order of their CPU utilization. The VM which tops this order of CPU utilization will be considered as a suitable candidate for migration. However, other metrics for this selection can be minimum migration time or largest memory image.

The live migration process: It starts with the profiling of every VM in a cluster while placing it on a physical machine by identifying which data-blocks are required at boot-time and also at application startup time. These data-blocks are collected and stored in the NAS device.

During the migration, the destination host copies this profiling information from the NAS device, without disturbing the execution of the VM at the source host. With the help of this information, destination host machine is able to reconstruct the full image of the migrated VM for booting up. Note that the complete image transfer of the VM has not yet taken place. This step helps in starting the VM at the destination host before the complete migration has taken place. This step is a significant improvement over the pre-copy approach where the new VM can start only after the full migration has taken place[4]. This also reduces the time for the new VM to resume execution once the complete image of the source VM is transferred.

Next, our method copies the execution log of the source VM up to the last checkpoint rather than copying the memory pages from source host machine to the destination host machine. This will not only decrease the data transfer volume but will also synchronize the destination VM with the source VM up to the last consistent state, without interrupting the normal execution of the source VM.

After the checkpoint file is transferred, the destination machine can start the application and the input-output requests can be redirected towards the destination VM. If, however, the requested data is not available with the new VM, then transfer of those dirty data-blocks is initiated from the source machine on a high priority basis[8]. After a while, the new VM will start functioning on an independent basis, and this will indicate that the old VM at the source host can be shut down.

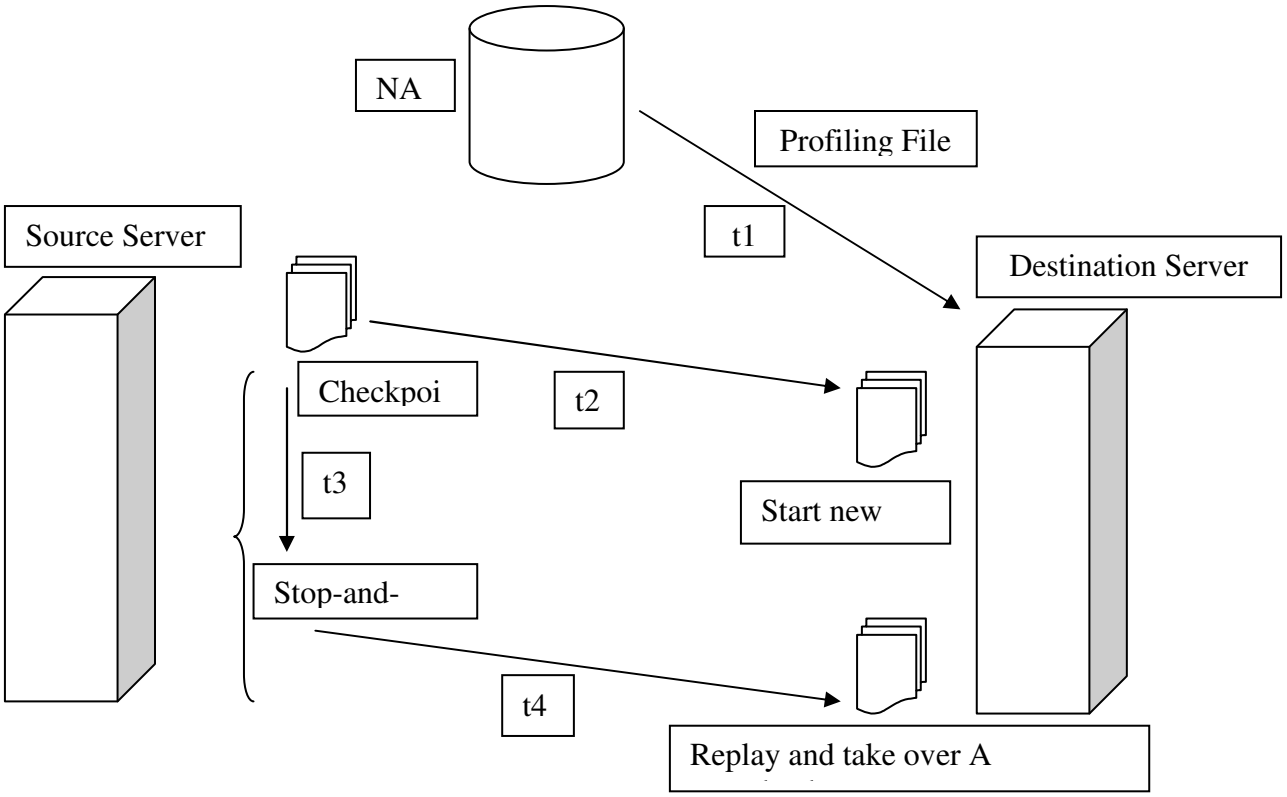

Figure-3
The detailed hybrid live migration process

Results and Discussion

Before discussing the performance of our hybrid live migration techniques, here are some important notations and their definitions:
R_{trans}- data transfer rate from source machine to destination machine. R_{trans1}- data transfer rate from NAS device to the destination machine. V_{prof}- size of the profiling file. V_{ckpt}- size of the checkpoint file. V_{dp}- size of the dirty pages after the last checkpoint

Here, the time to transfer the profiling file of the candidate VM from NAS device to the destination machine is depicted as

$$t_1 = V_{prof} / R_{trans1} \quad (1)$$

Next, the time to transfer the last checkpoint file from the source machine to the destination machine is

$$t_2 = V_{ckpt} / R_{trans} \quad (2)$$

The time to send the dirty pages from source machine to the receiving destination machine is

$$t_3 = V_{dp} / R_{trans} \quad (3)$$

Service downtime can be estimated as

$$t_4 = V_{dp} / R_{trans} \quad (4)$$

Total migration time, $T_{mig} = t_1 + t_2 + t_3 + t_4$ (5)
total data transmitted during migration, $V_{mig} = V_{prof} + V_{ckpt} + V_{dp}$ (6)

Figure 4 depicts the class diagram of our proposed migration procedure, consisting of 3 classes. Destination server and source server classes have 3 functions- monitor VM( ) which keeps track of a server's load, start VM( ) to boot a new VM on a server, and stop VM( ) to stop a running VM on a server. Similarly, NAS device allocates and de-allocates memory for each VM's profile file, also it initiates the transfer of profile file to any requesting server.

We have also mentioned the sequence diagram of our procedure in figure 5 below. It outlines the migration procedure of a VM from an overloaded server to another.

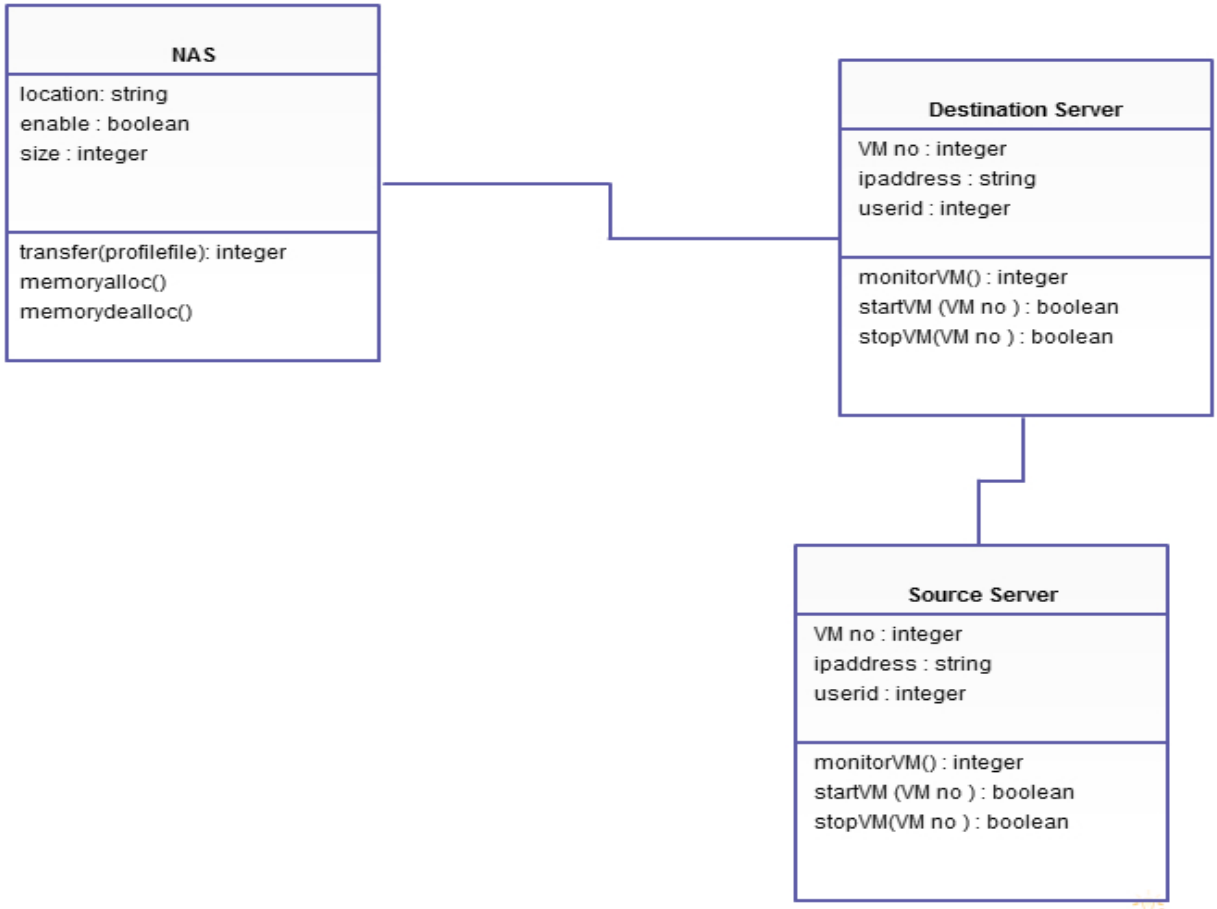

Figure-4
Class Diagram of hybrid Live VM Migration

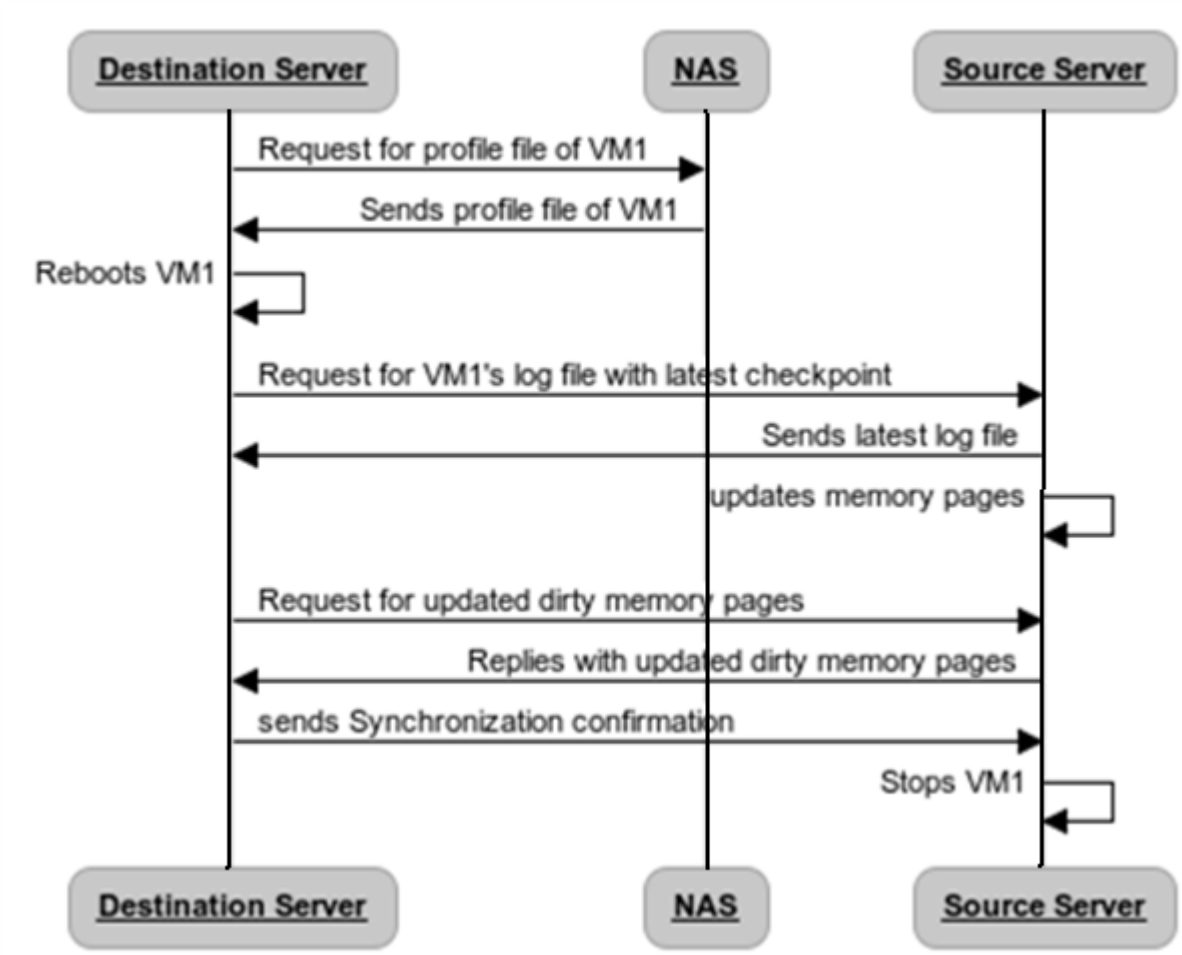

Figure-5
Sequence Diagram of hybrid Live VM Migration

Conclusion

In this paper, we have proposed a hybrid live VM migration process that combines the benefits of both the pre-copy and post-copy approaches. The proposed migration process is fast as there is no waiting for the complete VM image to be transferred before the destination VM can be started. By making use of the attached NAS device, less overhead is incurred in traffic-flow between the source and the destination hosts. Instead of copying the entire memory image of the VM, we are only transferring the execution log, there by further reducing the volume of data to be migrated. Lastly, taking advantage of the similar images of VMs of the same application, we have formed clusters of VMs so as to restrict the migration distance within the cluster itself.

References

1. Wood Timothy, Shenoy Prashant, Venkataramani Arun, and Yousif Mazin, Black-box and Gray-box Strategies for Virtual Machine Migration, 4th USENIX Symposium on Networked Systems Design USENIX Association & Implementation, (NSDI 07) **(2007)**

2. Strunk Anja and Dargie Waltenegus, Does Live Migration of Virtual Machines cost Energy?, *27th IEEE International Conference on Advanced Information Networking and Applications (AINA),* **(2013)**

3. Voorsluys William, Broberg James, Venugopal Srikumar, and Buyya Rajkumar, Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation, *Cloud Com '09 Proceedings of the 1st International Conference on Cloud Computing* **(2009)**

4. Perez-Botero Diego, A Brief Tutorial on Live Virtual Machine Migration From a Security Perspective, *Proceedings of the 2013 international workshop on Security in cloud* **(2013)**

5. Al-Kiswany Samer, Subhraveti Dinesh, Sarkar Prasenjit, Ripeanu Matei, VMFlock: Virtual Machine Co-Migration for the Cloud, *Proceedings of the 20th international symposium on High performance distributed computing* **(2011)**

6. Deng Wei, Liu Fangming, Jin Hai, Liao Xiaofei, 'Lifetime or Energy: Consolidating Servers with Reliability Control

in Virtualized Cloud Datacenters', *Cloud Computing Technology and Science (Cloud Com),* IEEE 4th International Conference **(2012)**

7. Jin Hai, Liu Haikun, Liao Xiaofei, Hu Liting, Li Peng, 'Live Migration of Virtual Machine Based on Full-System Trace and Replay', *18th ACM international symposium on High performance distributed computing* **(2009)**

8. Hines Michael R. and Gopalan Kartik, Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self -Ballooning, *VEE'09, March 11–13, , Washington, DC, USA*, **(2009)**

9. Boru Dejene, Kliazovich Dzmitry, Granelli Fabrizio, Bouvry Pascal and Zomaya Albert Y., Energy-Efficient Data Replication in Cloud Computing Datacenters, *Globecom 2013 Workshop - Cloud Computing Systems, Networks, and Applications* **(2013)**

10. Stage Alexander and Setzer Thomas, 'Network-aware migration control and scheduling of differentiated virtual machine workloads', *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing* **(2009)**

11. Gustafsson Erik, Optimizing Total Migration Time in Virtual Machine Live Migration, *www.diva-portal.org/smash/get/diva2:609356/FULLTEXT01.pdf* **(2013)**

12. Sun Meng, Gu Weidong, Zhang Xinchang, Shi Huiling, Zhang Wei, A Matrix Transformation Algorithm for Virtual Machine Placement in Cloud, *trustcom, 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom),* 1778-1783, (**2013)**

13. Hirofuchi T., Ogawa H., Nakada H., Itoh S. and Sekiguchi S., A Live Storage Migration Mechanism over Wan for Relocatable Virtual Machine Services on Clouds', *Proc. Ninth IEEE/ACM Int,l Symp. Cluster Computing and the Grid (CCGrid ',09),* 460-465, May **(2009)**

14. Khosravi A., Garg S. and Buyya R., Energy and Carbon-Efficient Placement of Virtual Machines in Distributed Cloud Data Centers, *Proc. 19th Int',l Conf. Parallel Processing (Euro-Par ',13), (***2013)**

15. Verma A., Ahuja P. and Neogi A., pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems, *Proc. ACM/IFIP/USENIX Ninth Int',l Middleware Conf. (Middleware',08),* 243-264, **(2008)**

16. Beloglazov A. and Buyya R., Energy Efficient Allocation of Virtual Machines in Cloud Data Centers, *Proc. 10th IEEE/ACM Int',l Symp. Cluster Computing and the Grid (CCGrid ',10),* 577-578, **(2010)**

17. Graubner P., Schmidt M. and Freisleben B., 'Energy-Efficient Virtual Machine Consolidation', *IT Professional,* **15(2),** 28-34, **(2013)**

18. Gandhi A., Gupta V., Harchol-Balter M. and Kozuch M.A., Optimality Analysis of Energy-Performance Trade-Off for Server Farm Management Performance Evaluation, **67(11),** 1155-1171, **(2010)**

19. Bobroff N., Kochut A. and Beaty K., Dynamic Placement of Virtual Machines for Managing SLA Violations, *Proc. 10th IFIP/IEEE Int',l Symp. Integrated Network Management (IM ',07),* **(2007)**

20. Ferreto T., Netto M., Calheiros R. and Rose C. De, Server Consolidation with Migration Control for Virtualized Data Centers, *Future Generation Computer Systems,* **27(8),** 1027-1034 **(2011)**