

## A Dynamically-adaptive Resource Aware Load Balancing Scheme for VM migrations in Datacenters

Dhruv Garg<sup>1</sup>, Kamal Kant<sup>2</sup>, Abhay Bansal<sup>3</sup>

<sup>1, 2, 3</sup>(Department of Computer Science and Engineering, Amity University, NOIDA, India)

---

**Abstract:** *The advancements in the field of networking, network infrastructures and the growing size for processing, storage/data and communication, the demand for Virtualization and Data Center flexible resource management has become significantly important. Virtualization is the single most effective way to reduce IT expenses while boosting efficiency and agility. And with time, the advancements in the field of Virtualization, increasing demands of datacenters and increasing network infrastructure development and maintenance cost lead to the development of Virtual Machines and their migrations across datacenters at large scale now. Though Virtual Machines in datacenter decouples physical machines with Virtual ones for their processing but they are still coupled with hosts for their virtual resources sharing and thereby exerting load on servers further. This paper focuses on balancing the load of the servers with the help of efficient VM migrations and proposes an algorithm that adapts to the dynamic business needs in a better way for datacenter access and request processing.*

**Keywords** – Virtualization, Virtual Machine, Migration, Hypervisor, Load Balancing

---

### I. INTRODUCTION

The advancements in the field of networking, network infrastructures and the growing size for processing, storage/data and communication, the demand for Virtualization and Data Center flexible resource management has become significantly important. A need arises for an approach that changes the economics of the data center by combining it into single unit for computation, storage, networking, virtualization, and management. And the end result has to be operational simplicity and business agility – essential for computations and IT deployments.

Virtualization is the single most effective way to reduce IT expenses while boosting efficiency and agility. And with time, the advancements in the field of Virtualization, increasing demands of datacenters and increasing network infrastructure development and maintenance cost lead to the development of Virtual Machines and migrations across datacenters at large scale now.

But with any advancement, there comes the loopholes in the new developments. Though Virtual Machines in datacenter decouples physical machines with Virtual ones for their processing to the end user at front end, but in back-end Virtual Machines are still coupled with hosts for their virtual resources sharing and thereby exerting load on servers further. This paper focuses on balancing the load of the servers with the help of efficient VM migrations and proposes an algorithm that adapts to the dynamic business needs in a better way for datacenter access and request processing. Section II of the paper will focus on the Virtual Machines and their migration strategies. Section III will discuss the issues and challenges related with VM migrations across datacenter followed with the proposed solution and simulation results in the further sections.

### II. VIRTUAL MACHINES AND THEIR MIGRATIONS

A virtual machine is a machine with its own virtual Operating System, configurations and virtually shared physical hardware. In other words, a VM is a replica or a copy of a real machine deployed virtually on a physical machine for its execution in view to dynamically map new systems with existing ones, increase number of users and system usage, test the new systems on existing infrastructure, and distribution and sharing of resources.

Though VM technology is not new in IT industry since IBM's mainframes designs in 1960's, but recent advancements in virtualization techniques made it possible to go further with movements of VM from one physical machine to another quite easy now. This came in advent to the cater growing demands, increase computational and processing needs, reduce maintenance costs and lower down the new network infrastructures deployment costs.

The term "migration" in IT networking and Datacenters, presents the movement of a system from one location or one network to another. VM Migration thus is a process performed between two or more physical servers deployed over a Local Area Network (LAN) or Wide Area Network (WAN) in which a Virtual Machine

is moved from one machines to another often in context to server maintenance, load balancing, efficient resource sharing, network scalability, testing, storage back-ups, server management, disaster recoveries, etc. And with the growing data day by day and advancements in application processing and networking, Virtual Machines and their migrations are becoming major needs of any business today.

## 2.1 Virtual Machine Architecture

Since a Virtual Machine comprises of its own Operating System and shared hardware, it resides independently, in isolation on a physical machine and executes over a Virtual Machine Monitor (VMM) or hypervisor. VMM is a Virtualization software designed to make virtual machines which constantly monitors the VMs running over it. It acts as a bridge between these VMs and physical machines' hardware/software.

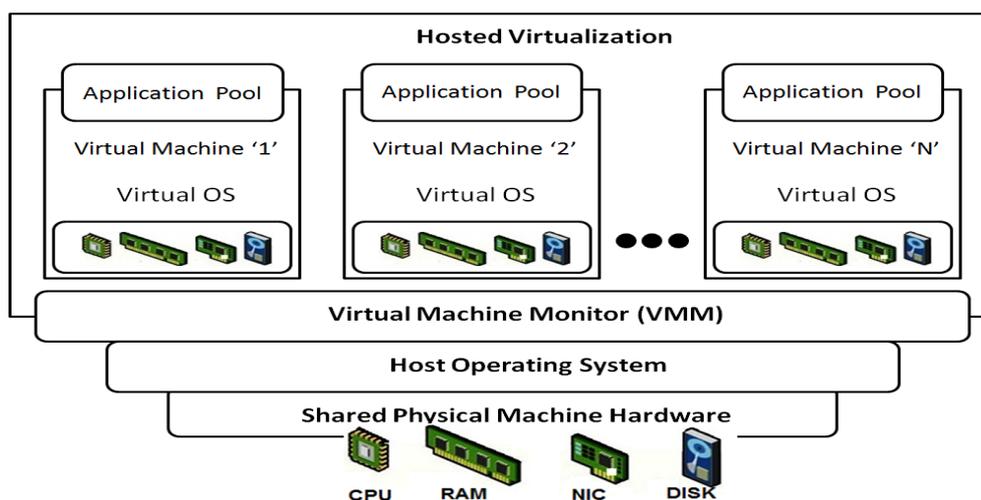


Fig. 1: Virtual Machine architecture for a server

## 2.2 Virtual Machine Monitors

A Virtual Machine Monitor (VMM, also known as hypervisor) is software that creates an abstract layer between Virtual Machines and host physical machines (PMs). This layer is designed to keep the VMs isolated from PM's OS and to allow sharing of physical resources by both PM and VMs residing over it. Hypervisors are also used to frequently check the VMs' states, loads and health. Hypervisors in the event of these features plays a very significant role during the VM migration process.

## 2.3 Virtual Machine Migration types

The VMs migration in datacenter is performed in two ways, firstly, the non-live VM motion. In this type VM to be migrated is suspended from all its operations, processing and is in shut-down state during whole movement process. The advantage of this migration technique is in easy movement of VM with its current state intact, without any user accessing it and no process in pipeline to get executed. But the major drawback as can be seen of this type of movement is non-operational VM state of machine been migrated. The users of that VM need to wait for undisclosed time span it takes for migration to re-access the VM again until migration is done. This may sometimes lead to violation of Service License Agreements (SLAs) and may result in huge business loss sometimes.

Alternatively, a second type of migration approach was developed named live VM migration with Pre-copy and Post-copy phases. In Pre-copy type the VM to be migrated remains in running operational state on the source machine while it's associated system configuration files, data files and storage is being migrated to the destination physical machine. Once the migration is complete, the VM is suspended for the source PM and resumed on target PM. The state changes and processing performed by VM while it was resumed on new PM are now transferred to it. These state transfers are called dirty pages. Surely it seems an alternative to non-live migration disadvantages but it also comes with its own drawback. Often highly accessed VM creates a huge amount of duplicate dirty pages during its migration process that the transfer of these pages after migration creates additional traffic and congestion to the existing network.

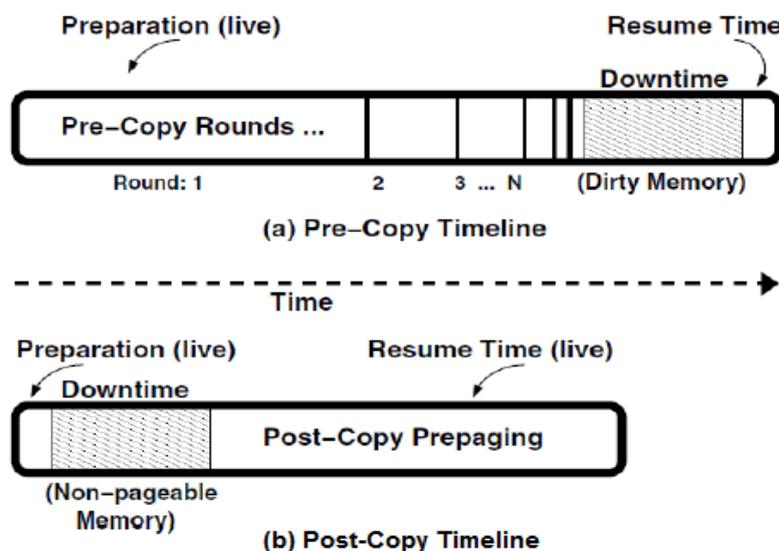


Fig. 2: (a) Pre-copy and (b) Post-copy migration strategies

Secondly, in Post-copy type the VM is suspended from the source machine before migration and transferred to target physical machine with minimum possible required configuration files for CPU cores and Memory configuration file and Storage memory to boot the VM on target host. After these transfers, VM start its execution on target physical machine while remaining associated on-demand memory is being transferred.

### III. VM MIGRATION ISSUES AND CHALLENGES

In the above sections we saw that the VM migration process takes place between two servers and in special cases multiple VM migrations processes can be performed simultaneously on more than two physical machines. Usually these VMs reside within one or more datacenters over a Local Area Network (LAN) but in some situations VMs can be migrated over a Wide Area Network (WAN) also. These machine transfers hence pose the following issues and challenges which still has a lot of scope to research upon:

- 3.1. *Latency*: Busy networks in datacenter often have high latency rates. So the successful smooth migrations of a VM within time need to have minimum possible latency between two servers in a datacenter network.
- 3.2. *Migrations with low impact factor*: Any VM mobility process performed must have the lowest affect to the end user's serviceability while accessing that VM.
- 3.3. *Network Bandwidth*: Every VM migration acquires certain amount of network bandwidth for its communication. High consumptions during this process may lead to network congestion and service degradation further.
- 3.4. *Machine Downtime*: To complete the migration a VM need to be suspended for reasonable time period. Long downtimes may result in high processing downtime, business losses, and user satisfaction degradation.
- 3.5. *Resource utilization*: To decrease infrastructure costs and increase user satisfaction levels for VM usability, the datacenter resources need to be utilized to their best performance levels using efficient and dynamic load balancing schemes. A good resource-aware migration scheme will always result in high usability of datacenter's network.
- 3.6. *Load Balancing*: Virtual Machines need to be constantly monitored and migrations need to initiate as soon as the scarcity of resources begins and load on the servers increases in datacenter. Efficient load balancing algorithm is required to deal with these issues.

### IV. JUSTIFICATION OF AREA OF STUDY

VM migrations across datacenters is a process of movement of VMs from one physical host to another in an intent to better utilize the physical resources of the datacenter and meet the growing IT business needs and demands. Machines could either be migrated to test new developments on existing networks, meet demands, disaster recoveries, storage backups, power savings, network maintenance or to balance the load of the physical hosts. These type of VM migration networks could easily find their existence in Storage Area Networks, IP-TV services, Cloud-based networks, etc. So it is necessary to understand the new technology in deep and enhance its usage further. Though different researchers did varied researches in the field since past decade to make efficient migration strategy, the development of a resource-aware migration approach is still to be answered.

## V. RELATED WORK

The increase in credibility of usage for datacenters and their virtualization in recent times have guided many researches in context to Virtual Machines and the migration process performance. Varied approaches and solutions were suggested by different researchers to resolve one or more issues listed in section 3.

Kento Sato et al.<sup>[1]</sup> proposed a model-based algorithm on data centric approach using VM based migrations in input/output based cloud applications. The focus of the research was to minimize the file access time by using the reallocation algorithm to choose the best strategy for migrating a VM to a location where target file is residing while reducing file access time while assuming that the network throughput between sites, the size and location of target file are given, which was a very weak assumption to migrate a VM from one host to another to improve network performance. The work required a lot of unnecessary computations for execution. Another approach for optimal VM migrations was proposed Sumit Kumar et al.<sup>[2]</sup> based on 'time-window' rather than 'event-based' migrations. The work stated the event-based migration i.e. initiating migration as soon as the CPU and Memory resources of VMs exceed that of PM capacity, as extremely myopic in nature. They instead proposed a migration procedure in which a global time-window (not application specific) is defined in which it allowed to list down the servers that presently or in future likely to be over-capacitated over a specific time span to arrive at the best migration map to minimize the adverse impact caused by such migrations<sup>[2]</sup>.

Ryousei Takana et al.<sup>[3]</sup> in their studied about the performance degradation in data intensive, high performance Input/output scientific computing environments which suffered with overheads of Virtualization and proposed a agent based solution named Symbiotic Virtualization technique for providing efficient migrations with Virtual Machine Monitor-bypass I/O technologies. This approach allows a hypervisor to work in association with a message passing layer on the virtual machine OS. Then with the help PCI hotplugs designed and hypervisors in the network, it chooses the optimum hypervisor on which the VM is to be migrated upon<sup>[3]</sup>. This work was mainly concerned with only fault tolerance in VM for using resource for long period of time with post copy migration schemes to hide overheads of hotplug and link-up by overlapping migrations.

Vivek Shrivastava et al. in their study worked upon inter-dependent homogenous-VMs migrations. They suggested that migrating a VM to a random physical machine will further increase the load on the network rather than lowering it. Instead they proposed a Application-aware migration strategy where VMs are moved to servers running similar application pool as executed by source VM. The work aimed to incorporating (1) inter-VM dependencies and (2) the underlying network topology into VM migration decisions<sup>[4]</sup>.

VMware in turn designed Distributed Resource Management (DRS) software to automate resource management with its previous distributions vMotion and vSphere<sup>[5]</sup>. DRS was designed keeping in view VM migration needs and reduce the related migration issues by using dynamic load balancing solutions for datacenter resources to meet VM demands all the time. It's related technology Dynamic Power Management (DPM) extends DRS capabilities further.

Another research [6] presented a Quality of Service framework for migrations for predicting the bandwidth requirements for migration process and predicts if the migration could be completed with specified time.

## VI. LOAD BALANCING ALGORITHMS

For the purpose of efficient VM migrations, many algorithms were designed according to the needs. Based on the state of the system and requirements, two types of algorithms Static and Dynamic Load Balancing were applied. Static algorithms like Round Robin, Central Manager, and Threshold algorithm assigns the tasks to the Virtual Machines and the assigned tasks ones assigned to a VM cannot be assigned to other machine to balance the load<sup>[16]</sup>. Dynamic algorithms are the ones that adapts with the changing requirements and states of the machines. These type algorithms allow the migration of a VM from over-loaded host to under-loaded host.

The proposed algorithm is thus based on the dynamic load balancing algorithm for efficient migrations of the VMs to balance the loads of the server. The algorithm is based on the awareness to the available resources in the datacenter like Memory and CPU utilizations of the servers at any instant of time. It is both event-based and time-window based algorithm which initiates the migration process as soon as any physical machine's CPU utilization goes above the specified threshold values. For this purpose the algorithm makes use of migrations test to find the optimal VM for migration and resource availability on target VM. The steps included in the algorithm are as follows:

- Step 1: Input the sets of Physical Machines (PM={P1, P2, P3,...}) and associated Virtual Machines (VM={V1, V2, V3,...}) along with their Memory and CPU utilization requirements.
- Step 2: Calculate average load (PMLoad\_average) of all the physical machines.
- Step 3: Generate the acceptance ranges  $\delta$  with the help of PMLoad\_average to divide the load arrays generated in step 1.

- Step 4: Make two sets (PM\_overloaded, PM\_underloaded  $\in$  PM) with help of  $\delta$  generated. Physical machines with loads above [PMLoad\_average +  $\delta$ ] are kept in PM\_overloaded set and PM loads below [PMLoad\_average -  $\delta$ ] are kept in PM\_underloaded.
- Step 5: Calculate a load variance,  $\sigma^2(M)$  of M physical machines to set the threshold value to set the migration process.
- Step 6: Now, if ( $\sigma^2(M) \leq \lambda$ ) then migration procedure exits. ' $\lambda$ ' is the datacenter server failure mark i.e. the maximum load limit a server can bear for its execution.
- Step 7: Now considering the PM\_overloaded and PM\_underloaded sets. Sort these sets in decreasing order of their load values.
- Step 8: Choose the heads of both the above sets i.e. the most over-loaded PM and Under-loaded PM.  $x = \text{HEAD}(\text{PM\_overloaded})$  and choose,  $y = \text{HEAD}(\text{PM\_underloaded})$ . Goto Step .
- Step 9: if (Migration = "SUCCESS")  
 remove (x)  
 set migration\_count= migration\_count+1  
 else  
 $x = \text{next} \leftarrow (\text{HEAD}(\text{PM\_overloaded}))$
- Step 10: if (migration\_count > size( $P_{OL}$ )) then goto Step 2.
- Step 11: Input 'x' and 'y' to the Migration\_Acceptance Test procedure.
- Step 12: Choose the highest load VM of 'x' chosen in step 8. Submit it for acceptance test with 'PM\_underloaded'.
- Step 13: For chosen VM in step above, perform Test 1 – Memory Test. In this test the selected VM is tested whether the Memory required by the VM to be migrated is available on the target server or not.

$$\text{i.e. } (\text{memory}(\text{VM}_{\text{new}}) \leq M_{\text{target}} - \sum_{i=1}^n \text{memory}(V_i)_{\text{target}},$$

where,  $M_{\text{target}}$  – maximum memory present in target machine, 'n' - no. of Virtual Machines, and  $\text{memory}(V_i)_{\text{target}}$  – memory utilized by VMs already residing on under-loaded PM.

- Step 14: then if the above test is a success, perform Test 2 – CPU Test.

$$\text{i.e. } \text{cpu}(\text{VM}_{\text{new}}) \leq \text{CPU}_{\text{target}} - \sum_{i=1}^n \text{cpu}(V_i)_{\text{target}} \text{ then}$$

- Step 15: if both tests are passed, initiate migration process from source to target PM. Set Migration = "SUCCESS". Remove (HEAD(PM\_overloaded)). Goto step 9.
- Step 16: if any of the above test is false, then choose another VM from PM\_overloaded selected. Goto step .
- Step 17: if after migration,  $\text{CPU}(\text{PM\_underloaded}) > [\text{PMLoad\_average} + \delta]$ ,  
 remove(HEAD(PM\_underloaded)).  
 Set  $y = \text{next} \leftarrow (\text{HEAD}(\text{PM\_overloaded}))$   
 Goto step 10.

The algorithm above provides with an equal opportunity to all the Over-loaded PMs to migrate their maximum loaded VMs to under-loaded PMs. Once any VM of a Over-loaded PM is migrated, it is removed and another Over-loaded PM is given the chance for migration of a VM to reduce its load. This way in one iteration of the complete algorithm, loads of all the servers in the datacenter are reduced significantly for their optimal performance. The acceptance ranges  $\delta$  is taken into consideration in order to reduce the frequent migrations with little variation of loads of the servers. The PMs within this acceptance range are considered to be normal loaded PMs which is the main aim of the proposed algorithm to bring the PM to normal state.

## VII. EXPERIMENTAL RESULTS

To prove the proposed algorithm mathematically, a Load balancing controller was designed in MATLAB software with variable loads sets of the servers in the datacenters. The designed controller was simulated for different datasets of CPU loads and Memory usages of the servers in the datacenter as represented below, for multiple times to generate result accuracy.

**Table 1: Dataset description – VM allocation scheme**

Physical Machine #	Virtual Machine #	Memory assigned (GB)	CPU assigned	VM Status
PM [1]	[1]	1.0	0.025	Active
	[2]	1.0	0.050	Active
	[3]	2.0	0.400	Active
	[4]	1.5	0.200	Active
PM [2]	[5]	0.5	0.090	Active
	[6]	1.0	0.100	Active
	[7]	1.0	0.150	Active
	[8]	2.0	0.200	Active
	[9]	1.5	0.250	Active
PM [3]	[10]	0.5	0.144	Active
	[11]	1.0	0.060	Active
	[12]	1.0	0.030	Active
PM [4]	[13]	2.0	0.200	Active
	[14]	1.5	0.110	Active
	[18]	0.5	0.0	Inactive
	[19]	1.0	0.0	Inactive
	[17]	1.0	0.0	Inactive
PM [5]	[15]	2.0	0.09	Active
	[16]	1.5	0.06	Active
	[20]	0.5	0.0	Inactive

**Table 2: Maximum CPU and Memory values available for Physical Machines**

Physical Machine #	PM_MaxMemory (GB)	PM_MaxMemory CPU
PM [1]	15.0	0.09
PM [2]	10.5	0.85
PM [3]	4.0	0.90
PM [4]	3.55	0.80
PM [5]	6	0.90

The above tables are one of the datasets used for the simulation purposes. Variable VM CPU load values were considered for result generations. VMs in datacenter in course of their simulation are activated and de-activated as the simulations progress to generate load variations for performance tests at variable loads. With the help of simulations conducted, following comparative graphs were generated for CPU utilization of the two stated algorithms in the previous section.

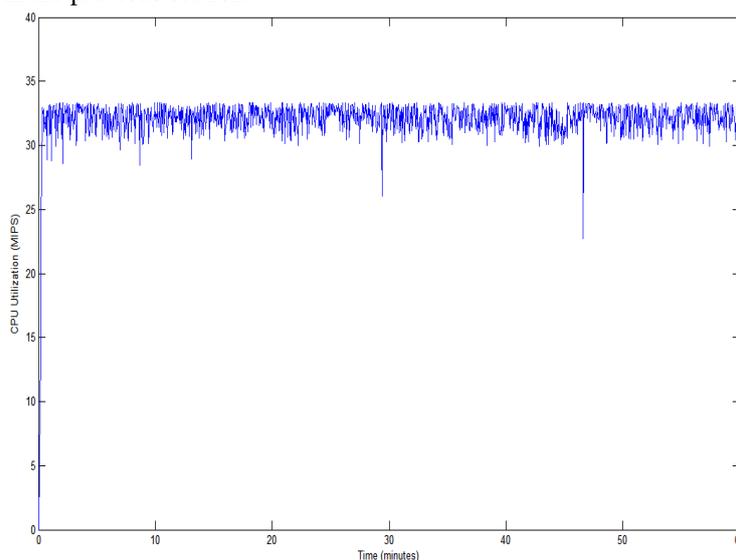


Fig. 3: Graph 1 shows overall CPU utilization MIPS of Datacenter PMs without load balancing algorithm

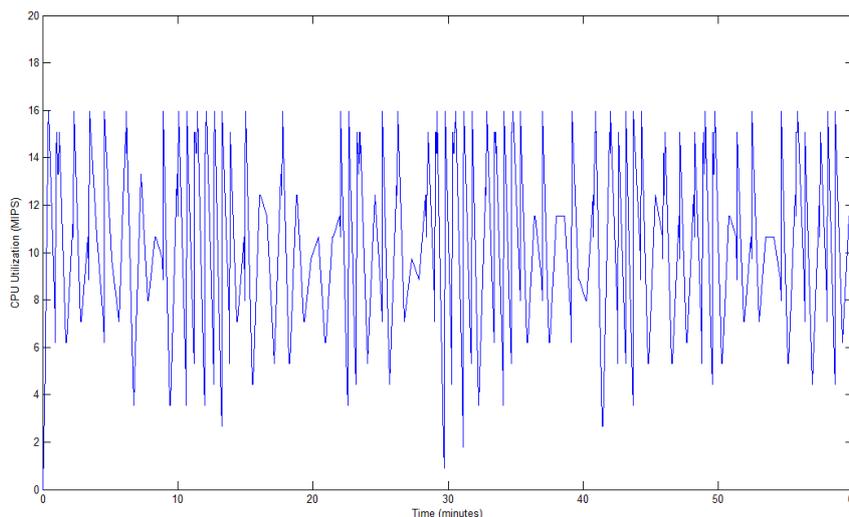


Fig. 4: Graph 2 shows overall CPU utilization MIPS of Datacenter PMs with load balancing algorithm proposed

The comparison graph 1 shows the CPU loads of the Round-robin algorithm used to allocate the resources to the VMs in the datacenters. Graph2 demonstrates the reduction in CPU loads after migrations with the help of proposed algorithm. Observations thus provide with following points after simulation for same sets of data:

- The CPU loads of VMs in datacenter are almost consistent with the round-robin algorithm for resource allocation after migrations.
- CPU loads became almost half after optimal migrations performed using the proposed algorithm.

## VIII. CONCLUSION

With the results generated through proposed algorithm it could be stated that an efficient, dynamically adaptive resource-aware load balancing algorithm like proposed above can significantly improve the overall efficiency and performance of the datacenter. This balancing of loads can make a better space for new Virtual Machines to be added to the datacenter and meet the growing business demands. Though the algorithm presents significant improvements in the loads of the datacenter, but to enhance the resource utilization of datacenter further, incorporating another resource test for bandwidth requirements of a VM to be migrated could predict the total time it may take to migrate the VM from source to target host as proposed by Vijay Mann et al. [6]. The proposed work in the paper could be easily deployed on datacenter networks of Cloud-based networks, IP-TV networks, for power savings in datacenters, etc. with minor modifications in the proposed work for both Homogeneous and Heterogeneous Virtual Machines.

## REFERENCES

- [1] Kento Sato, Hitoshi Sato, Satoshi Matsuoka, A Model-Based Algorithm for Optimizing I/O Intensive Applications in Clouds using VM-Based Migration, *Proc. ISCCG, IEEE*, 2009.
- [2] Sumit Kumar Bose, Srikanth Sundarajan, Optimizing Migration of Virtual Machines across Data-Centers, *Proc. ICPPW, IEEE* 2009.
- [3] Ryousei Takano, Hidemoto Nakada, Takahiro Hirofuchi, Yoshio Tanaka, Tomohiro Kudoh, Cooperative VM Migration for a Virtualized HPC Cluster with VMM-Bypass I/O devices, *IEEE*, 2012
- [4] Vivek Shrivastava, Petros Zerfos, Kang-won Lee et. al., Application-aware Virtual Machine Migration in Data Centers, *IEEE Infocom* 2011
- [5] VMware vSphere Storage Distributed Resource Scheduler, VMware White Paper, 2011 <http://www.vmware.com/products/datacenter-virtualization/vsphere/vsphere-storage-drs/features.html>
- [6] Vijay Mann, Anil Kumar V., Aakash I., Parantapa B., VMPatrol: Dynamic and Automated QoS for Virtual Machine Migrations, *IFIP, CNSM*, 2012
- [7] Weining Liu, Tao Fan, Live Migration of Virtual Machine Based on Recovery System and CPU scheduling, *IEEE*, 2011
- [8] Stefan Groesbrink, Basics of virtual machine migration on heterogeneous architectures for self-optimizing mechatronic systems, *Springer*, Oct. 2012
- [9] Zhenzhong Zhang, Limin Xiao, Mingfa Zhu, Li Ruan, Mvmotion: a metadata based virtual machine migration in cloud, *Springer*, Jan. 2013
- [10] Xiujie Feng, Jianxiong Tang, Xuan Luo, Yaohui Jin, A performance study of live VM migration technologies: VMotion vs XenMotion, *IEEE*, 2011
- [11] Prakash H.R., Anala M.R. and Dr. Shobha G, Performance Analysis of Transport Protocol During Live Migration of Virtual Machines, *IJCSE, Vol.2 5<sup>th</sup> ed.*, Oct-Nov 2011.
- [12] Yamuna Devi, Aruna P., Sudha Devi D., Priya N., Security in Virtual Machine Live Migration for KVM, *IEEE*, 2011.
- [13] Hines M., Gopalan, K., Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning., in *Proc. ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pp. 51–60, ACM, 2009.

- [14] (2012) The OpenVZ website. [Online]. Available: <http://www.openvz.org>
- [15] VMware vMotion: Live Migration for Virtual Machines without Service Interruption, 2009. <http://www.vmware.com/files/pdf/VMware-VMotion-DS-EN.pdf>
- [16] Aarti Khetan Vivek Bhushan Subhash Chand Gupta, A Novel Survey on Load Balancing in Cloud Computing, *International Journal of Engineering Research & Technology*, Vol. 2 Issue 2, February 2013
- [17] Varsha P. Patil, and GA Patil, Migrating Process and Virtual Machine in the Cloud: Load Balancing and Security Perspectives, *IJACSIT*, Vol. 1 Issue 1, pp. 11-19, 2012.
- [18] Mayank Mishra, Anwasha Das, Purushottam Kulkarni, and Anirudha Sahoo, Dynamic resource management using virtual machine migrations, *IEEE Communications*, vol. 50, pp. 34-40, September 2012..
- [19] Yuki Ashino, Masayuki Nakae, Virtual Machine Migration method between different Hypervisors implementations and evaluations, in *Proc. ICAINAW, IEEE*, 2012