International Journal of Advance Research in Science and Engineering
Volume No.08, Issue No.04, April 2019
www.ijarse.com
IJARSE
ISSN: 2319-8354

A NOVEL TECHNIQUE FOR LOAD BALANCING IN CLOUD ENVIRONMENT USING LIVE MIGRATION OF VIRTUAL MACHINES

Ashwini M. Bapat[1], Prof. M.V.Bramhe[2]

[1,2]Department of Computer Science and Engineering,

St. Vincent College of Engineering and Technology, Nagpur, Maharashtra,(India)

ABSTRACT:

Cloud Computing is becoming a promising paradigm in today's world of internet. Cloud computing offers computing resources on demand on a pay per use basis. This increasing demand of cloud computing causes overloading of organizations' data in the cloud which may result in Virtual Machine failure. Some datacenters are heavily loaded with the data while some datacenters are lightly loaded. This causes load imbalance. Hence, load balancing becomes an important task in cloud computing. Live virtual machine migration is used for this task. Most of the studies have successfully designed the techniques or algorithms for single virtual machine migration, multiple virtual machine migrations, load management, etc. Sometimes, unnecessary virtual machine migrations may occur. We have designed a load balancing technique which allocates Virtual Machines initially in order to prevent unnecessary virtual machine migrations.

KEYWORDS: *Cloud Computing, Migration, Virtualization, Virtual Machine, Load Management, Load Balancing, Virtual Machine Migration, Live Migration of Virtual Machines.*

1. INTRODUCTION:

CLOUD COMPUTING:

Cloud computing means the delivery of computing resources e.g., computer networks, servers, storage, applications & services over the internet on demand by pay per use basis. According to NIST, "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction."Cloud Computing provides three types of services. Software as a service, Infrastructure as a service, and Platform as a service.

VIRTUALIZATION:

We can create, recover data from cloud, we can host services, and we can even build our own apps and run them in the cloud. Due to these much benefits, the demand of cloud computing is increasing day by day. This increasing demand of cloud computing sometimes causes overloading of data in the cloud. This may cause Virtual Machine (VM) failure. This problem can be solved by virtualization. Virtualization technology plays an important role in the area of Cloud Computing. IBM introduced Virtualization technology in late 1960s. Most infrastructure as a

service (IaaS) providers use virtualization technologies, e.g. Xen & VMware, to deliver agile, scalable & low-cost infrastructures. It improves reliability and availability of resources. Through virtualization many Operating Systems run concurrently on a single physical host. Through virtualization physical machines can be divided into number of VMs.

VIRTUAL MACHINE:

A Virtual Machine (*VM*) is an operating system (OS) or application environment that is installed on software, which imitates dedicated hardware. Virtual Machine runs on host Operating System and provides virtual hardware to the guest Operating System. Guest Operating System runs on the Host Operating System. Each VM has its own Operating System.

NEED OF LOAD BALANCING:

Organizations are uploading their enormous data to the cloud. This huge data causes datacenters to be overloaded. This may cause VM failure. So, to avoid VM failure, poor application performance, to have the recovery of data, load balancing is necessary. Virtual Machine Migration (VMM) helps to achieve this goal. VMM is the process of transferring VM from one physical host to another physical host. VMM was first proposed by Clark at al.

LIVE MIGRATION OF VIRTUAL MACHINES:

Sometimes data needs to be migrated because if the datacenters are not available due to maintenance or security reasons clients will be disconnected. But clients should always be connected. Live Migration is used for this purpose. The task of migrating Virtual Machine (VM) from one physical host to another physical host in real time can be called as Live Migration of Virtual Machines. A user cannot detect an ongoing live migration of the VM. Figure 1 shows the process of Live Migration of Virtual Machines.

Live Migration is emerging as an outstanding technology to efficiently balance load. Cloud service providers use this technique. Migration of Virtual Machines prevents datacenters from overloading.

HYPERVISOR:

A hypervisor is also called as Virtual Machine Monitor which creates and runs Virtual Machines. A computer on which a hypervisor runs one or more VMs is called as Host Machine and each Virtual Machine is called as guest Machine.

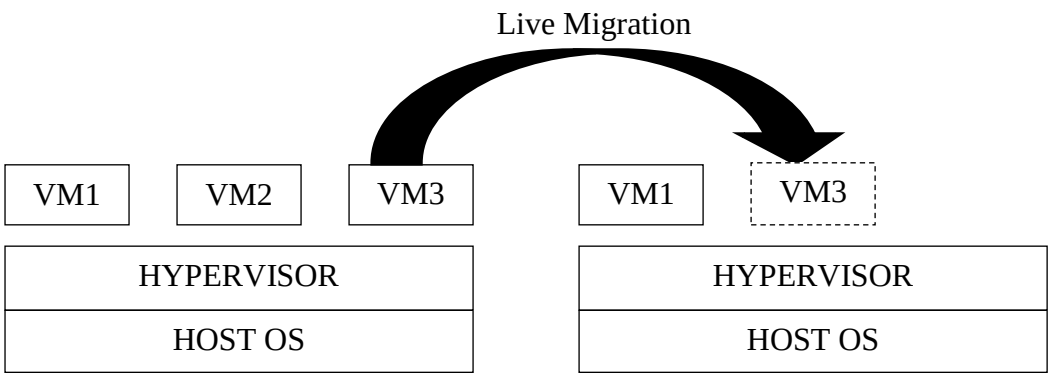

Figure 1.1: Live Migration of Virtual Machines

The remainder of this dissertation is organized as follows: Chapter 2 discusses the literature survey. Chapter 3 presents the identified problems and main objectives. Chapter 4 discusses proposed work and methodology. Chapter 5 presents implementation details. Chapter 6 presents results and analysis of implementation work. In Chapter 7 we conclude the research work.

2. LITERATURE SURVEY AND RELATED WORK:

This literature is a proof essay of the research work performed previously in the field of live migration of VMs for load balancing. It is the study and review of various drawbacks of existing system and various techniques for live migration of VMs.

The purpose of such a survey is to demonstrate the relevance of area of work and increase awareness on the relevant theories and the practices performed by previous authors, scholars on live migration of virtual machines. Furthermore, it is expected to be analytical of these experts or writer say and then agree or devise new method to add upon the existing work and continue the evaluation of various techniques in a way to enhance the quality and efficiency of live migration of virtual machines for load balancing.

Recently, there have been a number of studies on VM migration. In 2005, Clark et al. proposed the first live migration algorithm. Lalithabhinaya Mallu and Ezhilarasle R have surveyed relevant work in the area of Live Migration of Virtual Machines. Pre-copy live migration and Post-copy live migration are the two techniques of live migration of virtual machines [1]: In Pre-Copy Live Migration, A whole VM is sent to the destination and then modified memory pages are iteratively copied by the hypervisor. This process goes on till memory pages are copied to the destination host. In Post Copy Live Migration VM gets suspended on source and then memory pages are moved to the destination and then VM starts running on the destination regardless the whole pages have not been copied. The drawback of Post Copy live migration technique is Page faults may occur at the destination host. VM recovery is possible in pre-copy live migration because it keeps the copy of memory information on source machine. Pre-copy live migration is recommended in case of destination failure.

Authors in [2] proposed a Reliable Lazy Copy approach to Fast and Efficient Live Migration of Virtual Machines in the Clouds at a minimum cost. Lazy Copy approach helps to reduce total data transfer and downtime in pre-copy method and perceivable downtime in post-copy method. In the Push phase and Stop-and-Copy phase, in case of source failure, there is a permanent loss of VM. While in the case of destination failure there is no effect on the VMs. But in Pull phase, there is a loss of VM in both the cases. Hence, authors modified pull phase for making LC migration scheme to work in case of destination failures. The working is as follows: changes in the target host are reported to the secondary storage device. As the secondary storage device is accessible to both the source and target host, source gets updated with the changes reducing the recovery time of VM on source if there is a destination failure.

Authors in [3] designed the agent-based architecture for distributed load management. The load balancing protocol and an energy aware consolidation protocol are provided to the Agents to balance various loads while reducing energy consumption costs. When a host gets overloaded, agents work with each other to model resource usage of hosts and decide the best available target host for a VM. Agents can make the selection of the VMs to be

migrated and also, they may decide when and where to migrate, and when to turn on and off the hosts. Authors have focused on energy consumption cost reduction. Low overhead is the best advantage of this paper.

Gang Sun, Dan Liao, Dongcheng Zhao, Zichuan Xu, Hongfang Yu [4] have designed a VDC-M algorithm to migrate a Virtual Datacenter or multiple connected VMs among multiple datacenters for improving migration performance. The VDC migration (VDC-M) algorithm solves the problem of online migration for multiple VDC requests which arrive dynamically. The algorithm first remaps the VDC migration request, then finds migration paths and allots bandwidth resources for migrating virtual machines from the source servers to the destination servers. VDC-M algorithm also minimizes blocking ratio and VDC migrating cost. This algorithm also achieves low migrating cost and remapping cost. It minimizes the resource consumption because VMs can be possibly remapped to the same datacenter. This algorithm uses parallel migration strategy which results in lower downtime.

Authors in [5] have proposed a Hybrid Copy Live Migration to identify a reliable VM. In Hybrid Copy Live Migration, copy of memory is mapped from Source host to Destination host while VM runs on a Source host. Then, stopping the CPU & copy registers from source to destination host, whereas copying the CPU state the VM must be stopped, which causes clear traces which can be more easily traced. Finally, VM is removed from the source host and started on the destination host. Hence, VM runs on a new host. If the new host is not alike in software & hardware, these changes might be detected. Authors have designed a Hybrid approach for identifying the reliable VM: The user request is sent to all the available VMs. If a particular VM failed to receive a request it will not respond while all other active VMs will respond to the request. The current fault tolerance methods will find a reliable VM among all the VMs & then respond to the client request. Authors have proposed a willow path selection algorithm for data transmission from source to destination even if there are node failures during live migration of VMs.

Walter Cerroni [6] have presented a Markovian model for determining inter-Datacenter network performance of a federated cloud. The advantage using this model is smaller downtime.

To avoid potential violations of service-level-agreement (SLA) demanded by cloud applications, this paper [7] proposes iAware, a lightweight interference-aware VM live migration strategy. It captures the essential relationships between VM performance interference and key factors that are practically accessible through realistic experiments of benchmark workloads. iAware estimates and minimizes both migration and co-location interference among VMs, by designing a simple multi-resource demand-supply model. Extensive experiments and complementary large-scale simulations are conducted to validate the performance gain and runtime overhead of iAware in terms of I/O and network throughput, CPU consumption, and scalability, compared to the traditional interference-unaware VM migration approaches.

This paper [8] presents a migration optimization mechanism called *Clique Migration* for inter-cloud VM migration. *Clique Migration* divides a large group of VMs into subgroups based on the traffic affinities among VMs. Then, subgroups are migrated one at a time. First, the traffic monitor collects the traffic information between VM pairs. Then, the grouping mechanism profiles the traffic affinities, and makes a

migration decision to decide which VMs should be migrated simultaneously, as well as the order in which each subgroup of VMs should be migrated.

In this paper [9], authors have proposed a decentralized virtual machine migration approach inside the data centers for cloud computing environments. Authors have presented the design of the decentralized VM migration approach, which considers both load balancing and saving of energy costs by turning some underutilized nodes into sleeping state. The benefit of the decentralized approach is to eliminate the serious problem of single-point failure, which helps improve the availability of the entire system.

[10] In cloud architectures, users are almost not able to detect live migrations of their VMs nor can they prevent them from happening. Nevertheless, if a VM is live migrated to a distant data center crossing national borders, security & privacy problems arise. This way, internal data can become subject to new national legislation without even notifying the owner of the live-migrated VM. In this paper, authors proposed methods to detect live migrations from the inside of an affected VM. Furthermore, we analyze how the live migration procedure can be delayed & how the additional gained time can be used to take security measures before the live migration is finished. Authors developed a "live migration defense framework" (LMDF) which can be used for security policy enforcement within a VM.

A Virtual Machine (VM) is a software computer that runs an Operating System and applications [15]. VMs have virtual devices that provide same functionality as physical hardware [11]. Users can create VMs using VMware Workstations [11][12][13]. Users can specify no. of processors for VMs, allocate memory to VMs. The total amount of memory that can be assigned to all VMs running on a single host machine is limited only by the amount of RAM on the host machine [14]. VMware Workstations store standard VMs in a Virtual Machines directory [12].

3. IDENTIFIED PROBLEMS AND OBJECTIVES:

IDENTIFIED PROBLEMS:

Live migration is emerging as an outstanding technology to efficiently balance load. Virtual Machine Migration process lessens the overloading of data centers and provides the services without disruption. The main objective of migration process is to achieve less down time & migration time [1].

Various Live Machine Migration techniques are proposed previously for load balancing. From the study of aforementioned work following research gap has been identified. With the increasing demand of cloud services, organizations are uploading their data in a cloud. So, some datacenters are heavily loaded whereas some datacenters are lightly loaded. The solution for this problem is to provide efficient load balancing technique for cloud environment using Live Migration of Virtual Machine.

OBJECTIVES:

The objectives of this work are

- To design load balancing technique for allocating Virtual Machines initially to hosts in order to prevent unnecessary VM migrations.
- To test the proposed technique in cloud environment.

This proposed work is an attempt to address aims all or few of these problems mentioned above.

4. PROPOSED WORK AND METHODOLOGY:

In our work, we have used travelling salesman problem to show the live migration of virtual machines.

INSTANCE CREATION:

We have created 7 instances to show the initial allocation of VMs and the migration of virtual machines from heavily loaded node to lightly loaded node for load balancing.

The 7 instances are as follows:

i. Grid Server
ii. Grid Master
iii. Grid Node
iv. Grid Node 2
v. Grid Node 3
vi. Grid Node 4
vii. Client

PROPOSED LOAD BALANCING TECHNIQUE

The proposed load balancing technique is as follows:

STEP 1: Initialize the Grid Server, Grid Master, Grid Node and Client on Machine 1,
Grid Node 2 and Grid Node 3 on Machine 2,
Grid Node 4 on Machine 3.

STEP 2: Find the loads of Grid Node, Grid Node 2, Grid Node 3, Grid Node 4.

STEP 3: Allocate VMs to Grid Node 1, Grid Node 2, Grid Node 3, Grid Node 4.

STEP 4: Again Initialize the Grid Server, Grid Master, Grid Node and Client on Machine 1,
Grid Node 2 and Grid Node 3 on Machine 2,
Grid Node 4 on Machine 3.

STEP 5: Find the nodes which are heavily loaded and lightly loaded.

STEP 6: Perform the Live Migration of VM from overloaded nodes to the lightly loaded node.

STEP 7: Iterate from Step 4 to Step 6 until load is not balanced.

International Journal of Advance Research in Science and Engineering
Volume No.08, Issue No.04, April 2019
www.ijarse.com
IJARSE
ISSN: 2319-8354

This architecture of the proposed technique as follows:

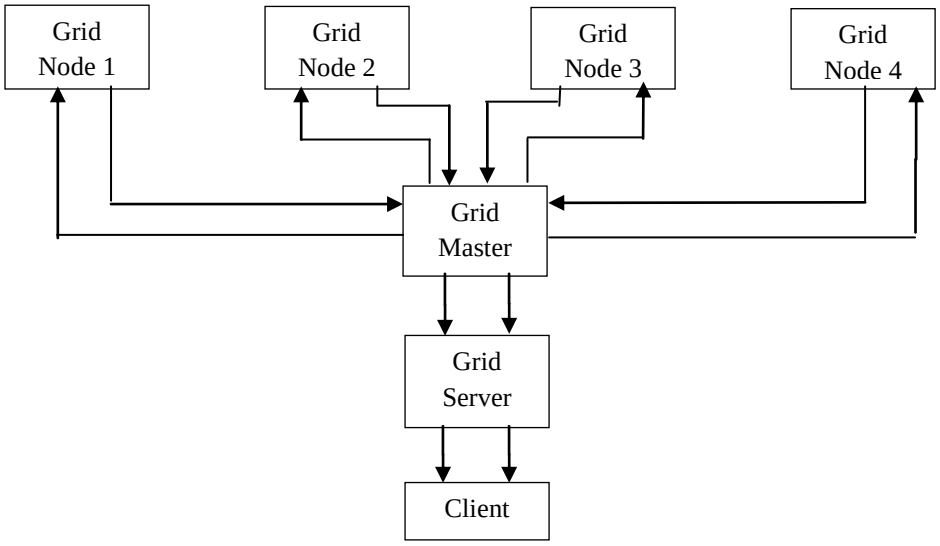

Figure 4.1: Architecture of Proposed Technique

Capacity of a node will be calculated as:

$C=\sum M + \sum P$

Capacity = Function of Memory left and Processing Power currently not in use

Live Migration of Virtual Machines:

In this scenario, consider the three machines Machine 1, Machine 2 and Machine 3. Machine 1 is 90% loaded and has 6 VMs, Machine 2 has 3 VMs and is 50% loaded and Machine 3 is 20% loaded and has 2 VMs. As we can clearly see in figure 4.2 Machine 1 is heavily loaded so it will send one VM to Machine 2 and two VMs to Machine 3.

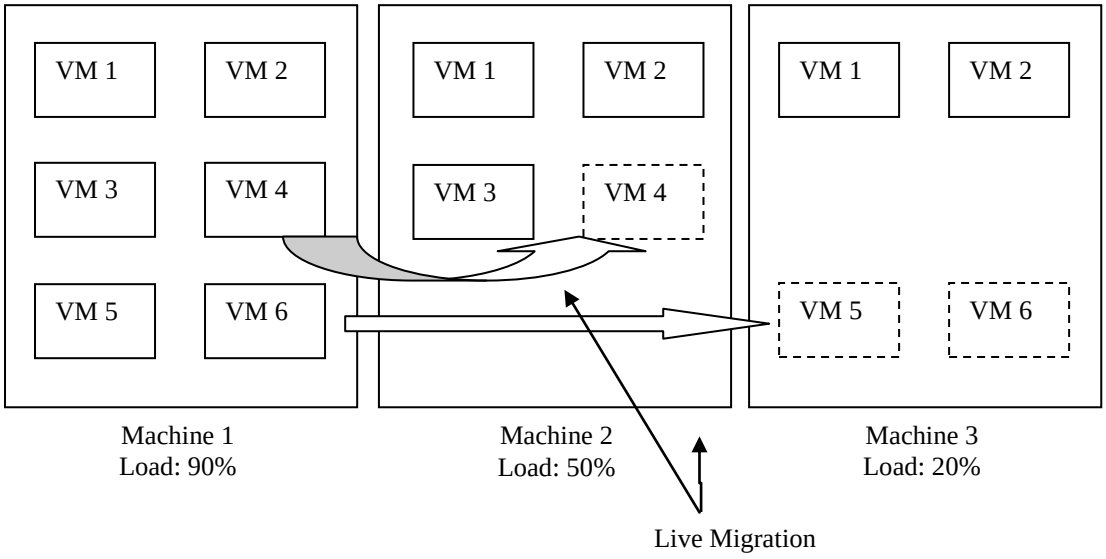

Figure 4.2: Live Migration Process

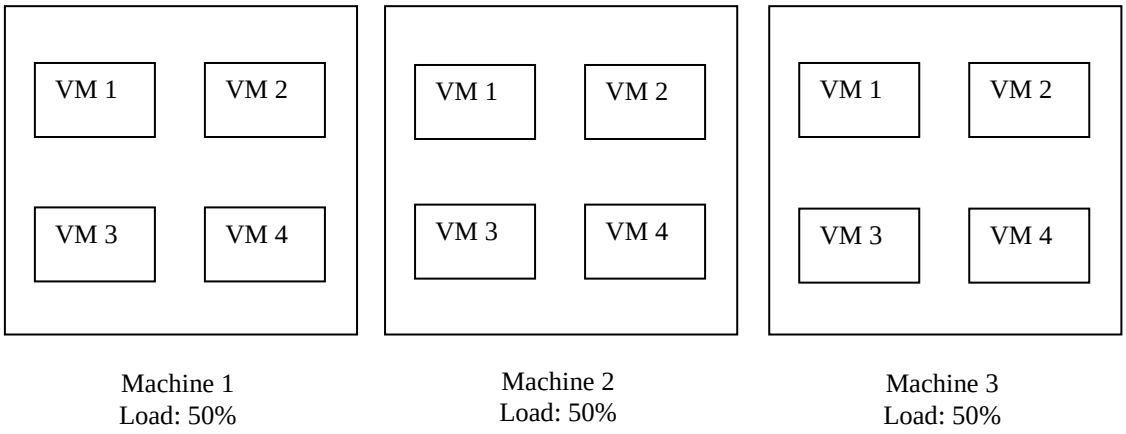

Figure 4.3: Condition of Machines after Live Migration

Figure 4.3 shows the condition of the machines after performing live migration. Here, we can clearly show how the load is balanced among the machines. After live migration, Machine 1 has 4 Virtual Machines making it 50% loaded, Machine 2 has 4 Virtual Machines & it is 50% loaded and Machine 3 also has 4 Virtual Machines & it is 50% loaded.

5. IMPLEMENTATION:

EXPERIMENTAL SETUP

In our work, we have considered three machines as private clouds and showed the live migration.

The specifications of the computers on which the execution is carried out are as follows: 1.80GHz Intel(R) Core(TM) i5-3337U, 6.00 GB (2.39GB usable) RAM, 32-bit OS, Windows 7, 1.70GHz Intel(R) Core(TM) i3-4010U, 2.00 GB (1.89 GB usable) RAM, 64-bit OS, Windows 8, 2.50 GHz Intel(R) Core(TM) i5-2450M, 4.00 GB (4 GB usable) RAM, 64-bit OS, Windows 7

Microsoft Visual Studio 2010, VMware 10

To show the Live Migration of Virtual Machines, we have used VMware Workstation 10 and Windows OS to create VMs.

IMPLEMENTATION DETAILS:

We have executed 4 instances on Machine 1 namely Grid Server, Grid Master, Grid Node and Client, 2 instances on Machine 2 namely Grid Node 2, Grid Node 3 and 1 instance namely Grid Node 4 on Machine 3.

Executing the code once will show the VM allocation and then subsequent executions will show the continuous VM migrations.

STEP 1: Initialization

Initialize Grid Server, Grid Master, Grid Node, Client on Machine 1, Grid Node 2, Grid Node 3 on Machine 2 & Grid Node 4 on Machine 3.

STEP 2:Execution of VM allocation

Initially, when the program is executed once, Client sends the request to Grid Server to solve the Travelling Salesman Problem. Grid Server further sends the request to the Grid Master. Grid Master sends the request seeking the available capacity of Grid Node, Grid Node 2, Grid Node 3 and Grid Node 4 for allocation of Virtual Machines (VM). The nodes send the reply to the Grid Master containing their available capacity. After knowing free capacity of all the nodes Grid Master decides the allocation of VMs. After the allocation of VMs, nodes individually solve the TSP and then send the result to the Grid Master. Grid Master receives the result from the nodes and then sends the result to Grid Server. Grid Server then sends this result to Client. Client shows the result of TSP.

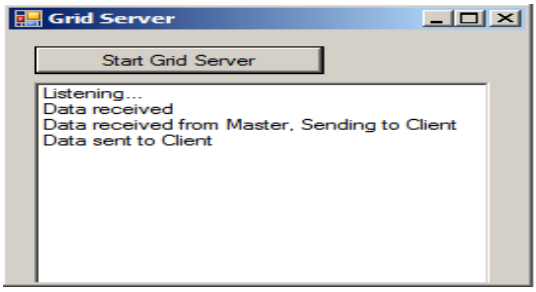

Figure 5.1: Screenshot of Grid Server during VM allocation

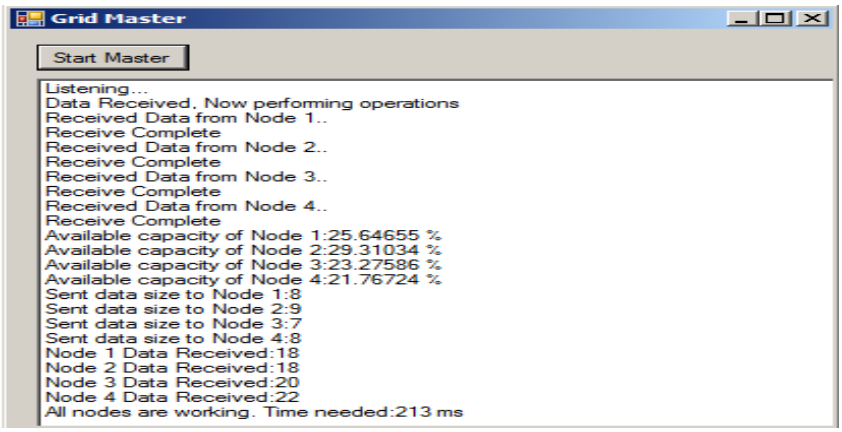

Figure 5.2: Screenshot of Grid Master during VM allocation

Figure 5.3: Screenshot of Client during VM allocation

STEP 3:Execution of VM migration for load management.

Here, Client again sends the request to Grid Server to solve the TSP. Grid Server further sends the request to the Grid Master. Grid Master sends the request seeking the available capacity of Grid Node, Grid Node 2, Grid Node 3 and Grid Node 4 for allocation of VMs. The nodes send the reply to the Grid Master containing their available capacity. After knowing the free capacity of all the nodes, Grid Master decides VM migration from heavily loaded nodes to lightly loaded nodes. Then again nodes individually solve the TSP and then send the result to Grid Master. Grid Master receives the result from the nodes & then sends the result to Grid Server. Grid Server then sends this result to Client. Client shows the result of TSP. Subsequent runs show the migration of VMs.

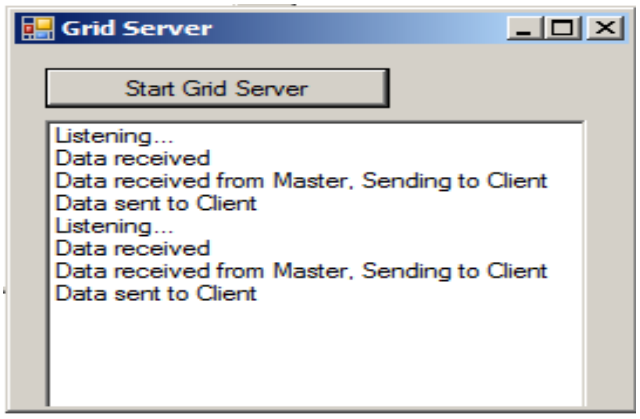

Figure 5.4: Screenshot of Grid Server after VM migrations

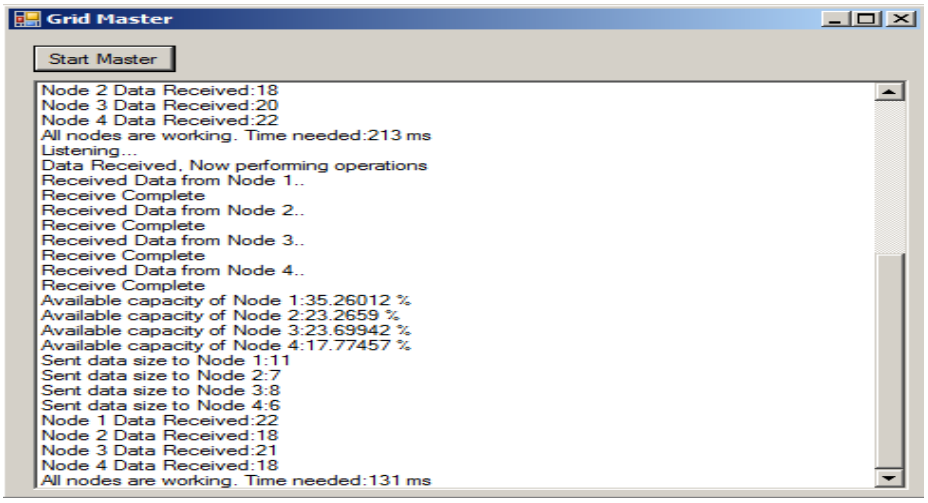

Figure 5.5:Screenshot of Grid Master after VM migration

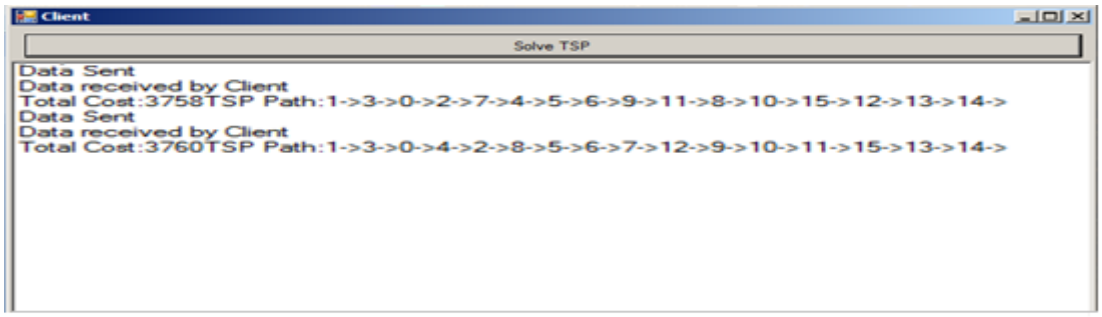

Figure 5.6: Screenshot of Client after VM migration

5. RESULT AND ANALYSIS

5.1 EVALUATION

Proposed technique is executed four times in Microsoft Visual Studio 2010. VMs are created using VMware workstation 10.

The specifications of the computers on which the execution is carried out are as follows: 1.80GHz Intel(R) Core(TM) i5-3337U, 6.00 GB (2.39GB usable) RAM, 32-bit OS, Windows 7, 2.70GHz Intel(R) Core(TM) i3-4010U, 2.00 GB (1.89 GB usable) RAM, 64-bit OS, Windows 8, 3.50 GHz Intel(R) Core(TM) i5-2450M, 4.00 GB (4 GB usable) RAM, 64-bit OS, Windows 7

In the figure 6.1 X- axis shows the Grid Nodes & Y-axis shows the Percentage of free capacity of Grid Nodes in the first execution. Graph shows the status of free capacity of Grid Node, Grid Node 2, Grid Node 3, Grid Node 4 during VM allocation.

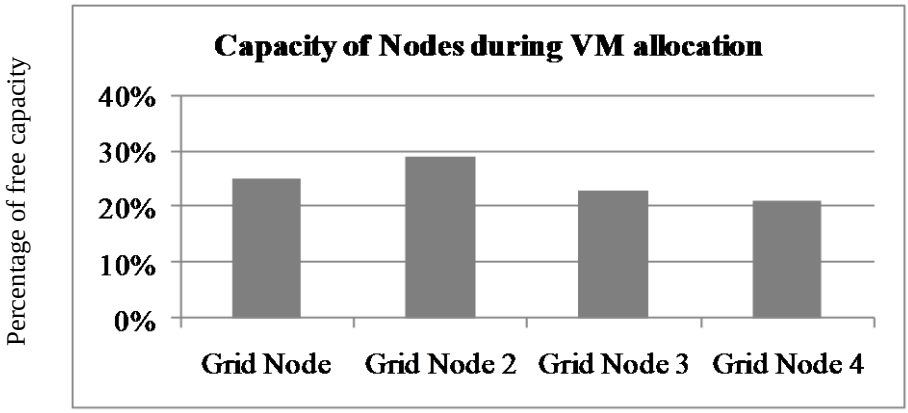

Figure 6.2 shows the status of Grid Node, Grid Node 2, Grid Node 3, Grid Node 4 during VM migrations carried out in second execution to balance the load. It can be clearly seen that Grid Node 4 is overloaded as it has only 17% capacity available. So 1 VM is migrated from Grid Node 4 to Grid Node 1 and Grid Node 4. X-axis shows the no. of Grid Nodes & Y-axis shows the percentage of free capacity of Grid Nodes.

Figure 6.1: Free capacity of Grid Nodes during VM allocation in 1st execution

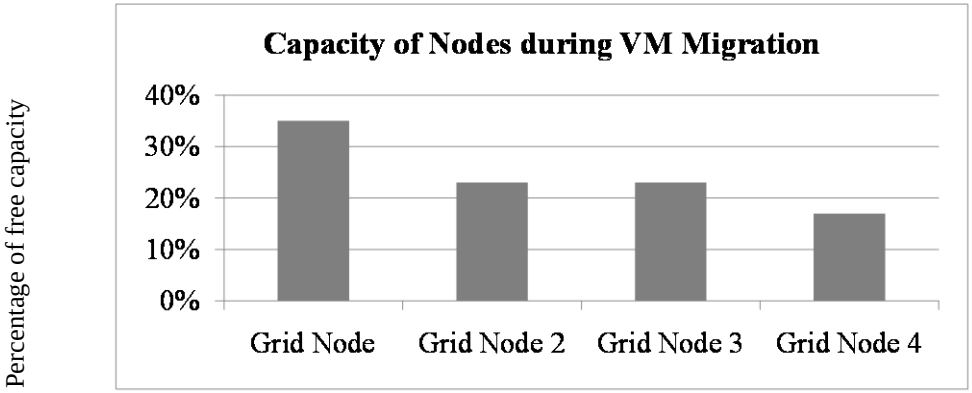

Figure 6.2: Free capacity of Grid Nodes during VM migration in 2nd execution

In figure 6.3, X-axis shows the no. of Grid Nodes and Y-axis shows the Percentage of free capacity of Grid Nodes. Graph shows the status of Grid Node, Grid Node 2, Grid Node 3, Grid Node 4 during VM migrations carried out in third execution. Grid Node is 39% free. Grid Node 2 is 25% free, Grid Node 3 is 20% free and Grid Node 4 is 15% free. Grid Node is lightly loaded as compared to all nodes. So 1 VM each is migrated from Grid Node 3 and Grid Node 4 to Grid Node. It shows how the load is balanced among all the Nodes.

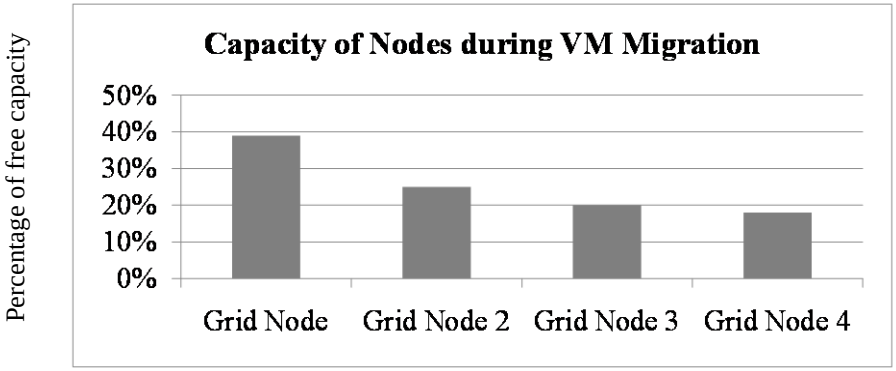

Figure 6.3: Free capacity of Grid Nodes during VM migration in 3rd execution

In figure 6.4, X-axis shows the no. of Grid Nodes and Y-axis shows the Percentage of free capacity of Grid Nodes. Graph shows the status of Grid Node, Grid Node 2, Grid Node 3, Grid Node 4 during VM migrations carried out in fourth execution. Grid Node is 25% free. Grid Node 2 is 20% free, Grid Node 3 is 25% free and Grid Node 4 is 30% free. Grid Node is lightly loaded as compared to all the nodes. So 1 VM is migrated from Grid Node 2 to Grid Node 4 to Grid Node. It shows how the load is balanced among all the Nodes.

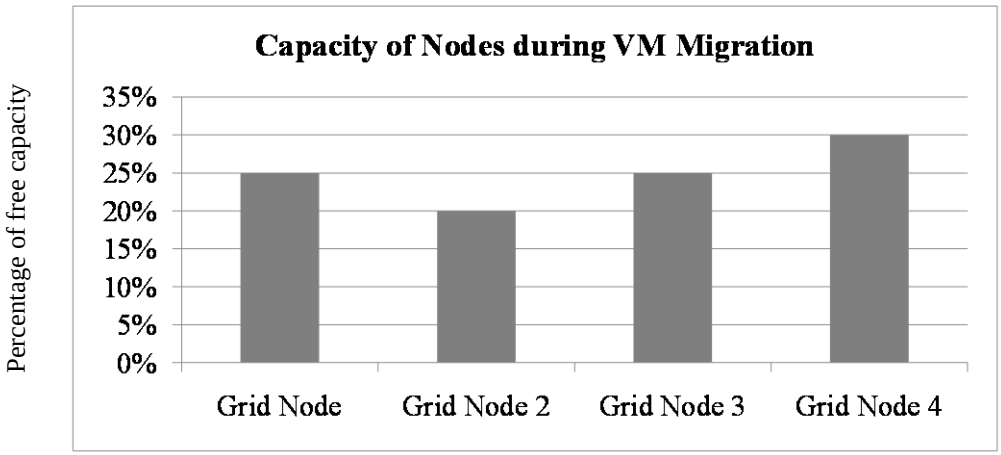

Figure 6.4: Free capacity of Grid Nodes during VM migration in 4th execution

6.2 ANALYSIS

Figure 6.5 shows the VM migrations among Grid Node, Grid Node 2, Grid Node 3, Grid Node 4. During 1st execution 4 VMs are allocated to all the Grid Node, Grid Node 2, Grid Node 3, Grid Node 4. In 2nd execution 1 VM is migrated from Grid Node 4 and no VMs are migrated from Grid Node 1, Grid Node 2 and Grid Node 3. In 3rd execution 1 VM is migrated from Grid Node 3 and Grid Node 4 each. X-axis shows the No. of Grid Nodes and Y-axis shows the No. of VM migrations. In 4th execution 1 VM is migrated from Grid Node 2. X-axis shows the No. of Grid Nodes and Y-axis shows the No. of Virtual Machines.

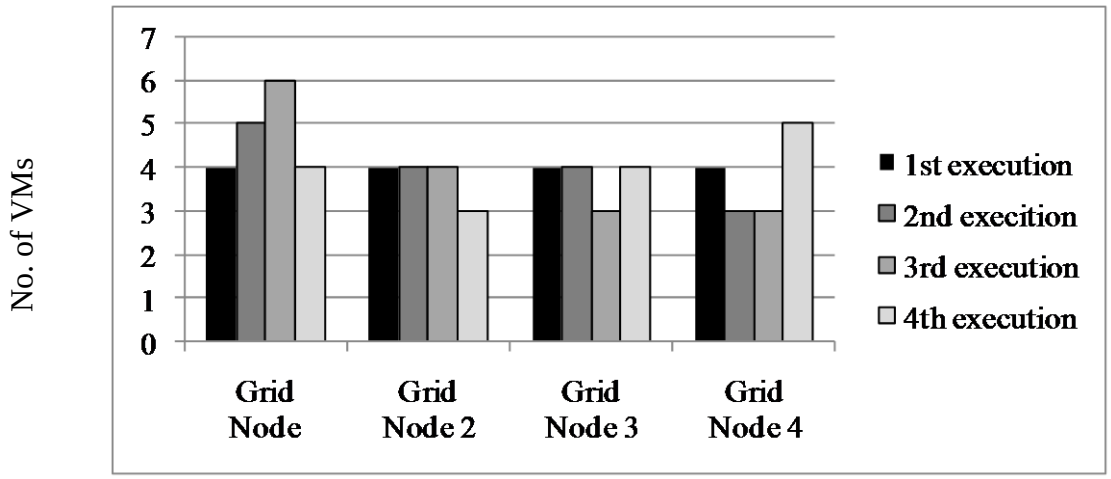

Figure 6.5: VM migrations among Grid Nodes

From this analysis, it is clearly seen that during 1st execution 4 VMs are allocated to all the Grid Nodes. During 2nd execution, only 1 Live Virtual Machine Migration is carried out. During 3rd execution only 2 Live Machine Migrations are carried out and in 4th execution only 1 Live Virtual Machine Migration is carried out.

So, our proposed technique prevents unnecessary Virtual Machine Migrations by carrying out very less Virtual Machine Migrations per execution.

7. CONCLUSION AND FUTURE WORK:

In this work we have proposed the load balancing technique using live migration of virtual machines. We have shown the initial allocation of Virtual Machine to the nodes by taking the available capacities of the nodes. Our proposed technique calculates the initial loads of the nodes to find the heavily loaded and lightly loaded nodes and then efficiently migrates the Virtual Machines from heavily loaded node to lightly loaded node which shows the load balancing among the machines.

We have tested this technique using three machines as private clouds. Our work is the successful implementation of a Live Migration of Virtual Machines which results in load balancing which in turn minimizes the node failure.

Our technique is executed on the private cloud infrastructure. The technique can be tested in public clouds. Some more virtual machines can be added to improve systems accuracy.

REFERENCES:

[1] LalithabhinayaMallu* and Ezhilarasie R, "Live Migration of Virtual Machines in Cloud Environment: A Survey", Indian Journal of Science and Technology, Vol 8(S9), 326–332, May 2015.

[2] Sanidhya Kashyap, Jaspal Singh Dhillon, Suresh Purini, "RLC - A Reliable approach to Fast and Efficient Live Migration of Virtual Machines in the Clouds", 2014 IEEE 7th International Conference on Cloud Computing.

[3] J. Octavio Gutierrez-Garcia and Adrian Ramirez-Nafarrate, "Collaborative Agents for Distributed Load Management in Cloud Data Centers using Live Migration of Virtual Machines", IEEE TRANSACTIONS ON SERVICES COMPUTING.

[4] Gang Sun, Yu Hongfang, "Live Migration for Multiple Correlated Virtual Machines in Cloud-based Data Centers", IEEE Transactions on Services Computing · January 2015.

[5] Miraculine.DV.M.Sivagami, "Efficient Data Transmission during Virtual Machine Failures Using Hybrid Copy Live Migration", Proceedings of National Conference on Communication and Informatics-2016, Organized by Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur.

[6] Walter Cerroni, "Network performance of multiple virtual machine live migration in cloud federations", Cerroni Journal of Internet Services and Applications.

[7] Fei Xu, Fangming Liu, Linghui Liu, Hai Jin, Bo Li, Baochun Li, "iAware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud", IEEE TRANSACTIONS ON COMPUTERS, VOL. 63, NO. 12, DECEMBER 2014

[8] Tao Lu, Morgan Stuart, Kun Tang, Xubin He, "*Clique Migration*: Affinity Grouping of Virtual Machines for Inter-Cloud Live Migration", 2014 9th IEEE International Conference on Networking, Architecture, and Storage

[9] Xiaoying Wang, Xiaojing Liu, Lihua Fan, and Xuhan Jia, "A Decentralized Virtual Machine Migration Approach of Data Centers for Cloud Computing", Hindawi Publishing Corporation Mathematical Problems in Engineering, Volume 2013, Article ID 878542.

[10] Sebastian Biedermann, Martin Zittel and Stefan Katzenbeisser, "Improving Security of Virtual Machines during LiveMigrations", 2013 Eleventh Annual Conference on Privacy, Security and Trust (PST)

[11] https://www.vmware.com/pdf/desktop/ws1001-using.pdf

[12] http://www.vmwarebits.com/Creating-your-first-virtual-machine-in-VMware-Workstation

[13] https://www.slideshare.net/marciniwinski/creating-a-new-vm-in-vmware-workstation

[14] https://pubs.vmware.com/workstation-10/index.jsp

[15] Thomas Erl, Cloud Computing: Concepts, Technology and Architecture (2013)

[16] Sandeep Bhargava, Swati Goyal(Suresh Gyan Vihar University,Jaipur), "Dynamic Load Balancing in Cloud Using Live Migration of Virtual Machine ", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2 Issue 8, August 2013