Network Bandwidth Aware Dynamic Automated Framework for Virtual Machine Live Migration in Cloud Environments

Surath Liyanage

A thesis submitted in partial fulfilment of the requirements of Kingston University for the degree of Doctor of Philosophy

Faculty of Science, Engineering and Computing
Kingston University

June 2018

Abstract

Live migration is a very important feature of virtualisation, a running VM can be seamlessly moved between different physical hosts. The source VM's CPU state, storage, memory and network resources can be completely moved to a target host without disrupting the users or running applications. Live VM migration is an extremely powerful tool in many key scenarios such as load balancing, online maintenance, proactive fault tolerance and power management.

There are four steps involved in the live VM migration, the setup stage, memory transfer stage, VM storage transfer stage and the network clean up stage. The most important part of live VM migration is transferring the main memory state of the VM from the source to the destination host which can consume a significant amount of network bandwidth in a short period of time.

Modern cloud based data centres generate a significant amount of network traffic apart from VM live migration traffic. If VM migration occurs during a peak time, VM migration and user traffic will compete for network bandwidth, then the data centre's network may not have enough resources to support both VM migration and demands of application users, which would create a bottleneck in the network. Therefore, this research presents a centralised, bandwidth aware, dynamic, and automated framework for live VM migration in Cloud environments. The proposed framework adopted a heuristic approach, and it provides guaranteed bandwidth for VM live migration by controlling user traffic on the network while scheduling live VM migration in an efficient manner.

The framework consists with two main components, The Central Controller and the Local Controller. The Local Controller is responsible for collecting resources usage data from VMs and PMs however the Central Controller makes global management decisions. The Central Controller is based on four algorithms which are called a migration policy. The migration policy contains the following algorithms: the host overloaded detection, host underloaded detection, VM selection and VM placement algorithms which are proposed in this research. The proposed migration policy has been implemented in CloudSim and evaluated against two benchmark migration policies in CloudSim. Five evaluation metrics have been used in the simulation to evaluate the performance of the proposed migration policy. The results reveal that the proposed migration policy outperformed the two benchmark policies.

Acknowledgments

PhD is a once-in-a life time opportunity and experience. It has been a period of intense learning for me, not only in the scientific arena, but also on a personal level. I am truly happy that I have had a chance to complete it.

I would like to express my special appreciation and gratitude to my supervisor Professor Souhel Khaddaj who has given me the opportunity to undertake a PhD and provided with invaluable guidance and advice throughout my PhD candidature. You have inspired me to achieve at a level higher than I thought possible. I would like to express my gratitude to the PhD committee members for their constrictive feedback and suggestions on improving my work.

I would also like to thank Dr: Mahdi Aiash, a lecturer at Middlesex University where I did my MSc. He has been helpful in providing guidance on finding a topic and preparing the project proposal for the project.

I am heartily thankful to my parents my father, S.Liyanage and my mother, Seetha Wickramasinghe for encouraging me in all of my pursuits and inspiring me to follow my dreams. I thank my sister Nisansala and sister –in –law Nicole for helping me with childcare, which helped me to work on my PhD without stress. And I also thank our family friend Caroline for helping me with proofreading, I think of her as a big sister. I would like to thank my daughter Saveinya Liyanage, who is two and half years old at the time of writing the thesis, sometimes she had sacrificed her play time with Daddy and shown great amount of patience when I was working on the project.

Finally, I would specially like to thank my wife Shehani who has been extremely supportive of me throughout this entire process and has made countless sacrifices to help me get to this point. I could not have completed my research without your support.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

Cloud Computing has revolutionised the IT industry in recent years. It is a paradigm where computer processing, storage, and network capacity are made available to users in an on demand manner through virtualisation on a shared physical infrastructure (Trieu et al. 2010). The Cloud Computing Concepts are based on distributed, parallel and grid computing coupled with virtualisation.

There are three basic service modules in the Cloud Computing, Software as a Service (SaaS), Platform as a Service (Paas) and Infrastructure as a Service (IaaS). Today many organisations are experiencing the benefits of Cloud Computing; they built out Private and Public Clouds using various commercial cloud providers such as VMware, Microsoft Azure or open source platform such as OpenStack. They are already establishing online services that are not limited to internal users, but outside their firewalls as well.

Today's internet applications can take the advantage of the Infrastructure as a Service (IaaS) model in Cloud. There are many IasS providers, for instance; Amazon reports several case studies that leverage their EC2 platform, including video processing, generic simulations and web applications. Such platforms are particularly useful for multi-tier web applications such as Apache, an application server / dynamic content generation, such as PHP and Java EE and backend databases such as MySQL and Oracle. Social networking sites have gained popularity in recent years and they are the most notable example of highly dynamic and interactive Web 2.0 applications. The increasing popularity has made demand for highly scalable, reliable and flexible solutions for hosting applications. These web applications have additional features that make them different from traditional static workloads (Voorsluys et al. 2009), and Cloud Computing perfectly suits their dynamic demands.

Virtualisation is the key technology behind Cloud computing that supports simultaneous execution of diverse tasks over a shared hardware platform. The Virtual Machine (VM) is a software implementation of a computing environment in which an operating system or program can be installed and run (Kapil et al.2013). In a Cloud Computing environment, applications and services are hosted on Virtual Machines that span over several physical

servers with dedicated resources (CPU cores, RAM, Disk Space, etc) which are allocated to each VM in order to closely match the applications' needs. Virtualisation provides many benefits, such as resource utilisation, portability, application isolation, reliability, higher performance, improved manageability and fault tolerance (Kapil et al.2013)

Resourceful delivery of IT services requires a highly scalable and smart network infrastructure that provides end-to-end delivery of IT services. Both users and service providers agree that the network infrastructure is the most critical and important asset in deploying and delivering services to users. All these concepts mentioned earlier play a fundamental role in building a cloud computing infrastructure, which provides reliable and resilient service delivery to users.

Cloud Computing can be used to build large computing infrastructure for large scale data centres, which are equipped with hundreds or thousands of physical nodes with multiple virtual machines running on them. A typical large cloud data centre consists of a large number of hosted servers and runs thousands of Virtual Machines (VM), these physical servers or machines (PM) are connected with high speed communication links. There are different types of data centres, which are built to serve different purposes. Data centres can be categorised into three main types (Benson et al.2010).

- University data centres
- Private enterprise data centres
- Commercial data centres

University data centres serve the students and administrative staff of the university. They provide a variety of services including Email services, Web services, system back-ups and multicast video streaming. These types of datacentres evolved over time, moving from a collection of devices in a storage closet to a dedicated room for servers and network devices.
Private enterprise data centres have been implemented to serve corporate users, developers and their small number of customers according to the company's requirements. They support a large number of custom applications and development test beds in addition to traditional services such as Email, Storage and Web services.

Commercial cloud data centres, unlike the first two types of data centres are built to serve a wide variety of cloud users. These types of data centres provide different types of cloud services in addition to traditional services including Internet-facing services, search engines,

indexing, social media, video and data mining. Most importantly, commercial cloud data centres are purposely built to serve external cloud customers and they can use its services by paying a subscription or using the pay-as-you-go model.

A cloud data centre might be equipped with hundreds of physical servers that host thousands of Virtual Machines (VM). Those VMs which are operating in data centres can be migrated across different physical nodes on demand to achieve various goals, and this is known as VM migration. There are two types of VM migration categories, hot (live) or cold VM migration. In live or hot VM migration, a VM can move a powered on virtual machine to a different host without any interruption in the availability of the virtual machine. Cold VM migration is involves moving a powered off or suspended VM to a new host (Eramo et al.2017).

Live migration is a very important feature of virtualisation, a running VM can be seamlessly moved between different physical hosts. Source VM's CPU state, storage, memory and network resources can be completely moved to a target host without disrupting the users or running applications. Live VM migration brings many benefits to a cloud provider's environment. Live VM migration can be used in order to improve performance, the system's resources utilisation and power utilisation. As an example, if a server is overloaded, some VMs can be moved to underloaded servers in order to achieve energy efficiency and reliability.

Unplanned downtime of a cloud service may occur due to the failure of a physical server, which will cause SLA violation with critical business applications running on hosting virtual machines. Reliability can be achieved by migrating VMs from a failing or underperforming physical server to a working physical host server. It's important to minimize unplanned downtime and also recovery time of technical problems in order to avoid disastrous situations. Recently, most virtualisation products offer seamless live VM migration that involves extremely short downtimes ranging from tens of milliseconds to a second.

However, live VM migration can consume a significant amount of network bandwidth for several seconds (500 Mbps for 10 seconds for a trivial web server which hosts on a VM), so these non-negotiable overheads need to be considered when scheduling migration (Stage et al.2009). Higher workload density of VMs in combination with intensive VM migration can lead to network congestion if there is not enough bandwidth to support both higher workloads and VM migration. To secure the bandwidth for VM migration, it is necessary to have a mechanism to schedule VM migration and also allocate necessary bandwidth for VM

migration. If we can find a mechanism to predict the bandwidth which is required for VM migration prior to the migration, as well as the current user traffic and also the available bandwidth on the network, then that information can be used to schedule VM migration in order to avoid a network bottleneck. If there is not enough bandwidth on the network to support VM migration, limiting the user traffic on busy physical machines will facilitate allocation of the necessary bandwidth for VM migration and will guarantee the minimum bandwidth for VM migration. In order to address the problems that are associated with virtual machine migration, a new framework is proposed and discussed in chapter 5.

1.2 Research Problem Statement

1.2.1 Live VM Migration Overhead

In live VM migration there are certain parameters of source VM that need to be transferred to the destination host, which are,

- VM's virtual memory
- VM's storage
- Network state

As most data centres today use share storage within the migration cluster, the VM's storage does not need to be transferred to the destination. The file system state consistency during live VM migration within a same cluster is typically ensured by adopting shared storage solutions, such as Network Attached Storage (NAS) and Storage Area Network (SAN) (Network Evolution, 2011). In this research, it is assumed that the migrating VMs are attached to the same file system, available at both source and destination host, so that there is no need to copy disk images.

The network state transfer issue is easily solved in a local cloud environment since each VM is connected via a virtual switch to the same physical LAN (Local Area Network) at both source and destination hosts. It will keep the same IP address, when the execution is resumed at the destination and send a ARP packet to all switches and neighbours, which will be aware of the new VM location . A more complex scenario occurs when migrating a VM to a host that is connected to a different IP network, but this scenario is out of the research scope.

The most important part of live VM migration is transferring the main memory state of the VM from the source to the destination host. This research only focuses on transferring the

main memory state of the VM's memory from source to destination host, which is the most difficult process in live VM migration. Recently a large VM memory was being used, for example, Amazon EC2 provides several 8xlarge in stance types with 244 GiB of memory (Suetake et al.2016) . Transferring large VM memory content will increase migration time and lengthy migration prolongs the resource pressure at the source, which in turn degrades the performance of all VMs and also the PM plus it creates network bottlenecks.

During the live VM migration process, in both memory transferring techniques (Pre-copy and post-copy), it is required to transfer all VM memory during VM migration, not just the working set. However, any swapped out memory pages of the migrating VM need to be swapped back in before being transferred (Deshpande et al.2016).

What is a swapped out Page?

Hypervisors can use a sledgehammer approach to virtual memory management which involves swapping VM physical memory (and the associated host physical memory) directly to a disk. Generally, in VM migration a swap space for each powered-on Virtual Machine is created but powered-off VMs do not require a swap space. There is one swap file for each powered-on VM, and the size of a swap file (Figure 1.1) can be calculated using the following formula.

Swap file size = Configured VM memory – memory reservation

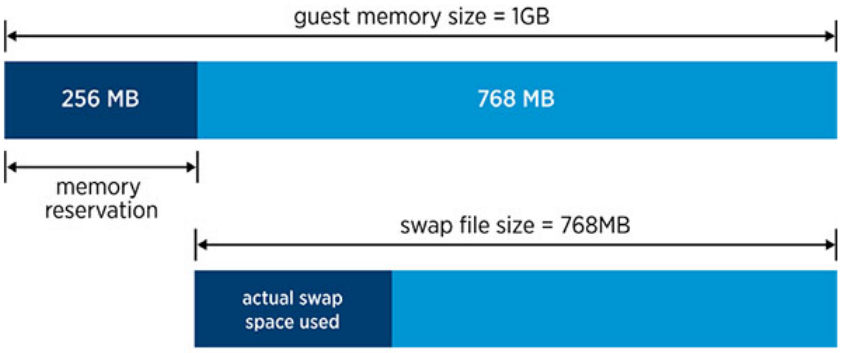


Figure 1.1: Swap Space, the VM has a configured memory of 1GB and a configured memory reservation is 256 MB. (Banerjee et al.2014)

The figure 1.2 shows the swap pages transferring process in both pre-copy and post-copy VM migration techniques. The Migration Manager swaps all pages swapped out from the source host, and then sends them through a direct TCP connection to the destination.

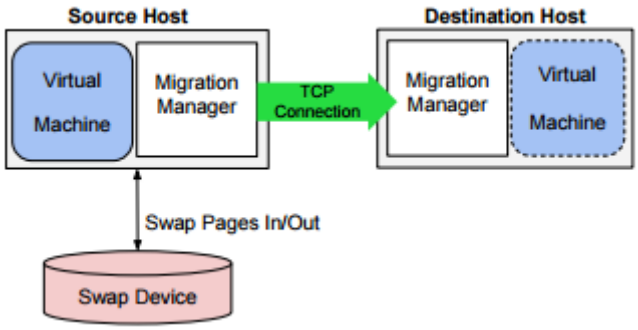


Figure 1.2: Swap Pages In/Out (Deshpande et al.2016)

The Migration Manager is an external management process that associates with each VM and carries out the VM migration. The Migration Manager at the source host establishes a TCP connection with the Migration Manager at the destination host, and then transfers the VM's CPU state, VM memory and I/O device states.

When a VM memory migrates from a host to the destination, the host pulls the swap pages back in to the memory and they are transferred to the destination host. As the pages are copied, the swapped pages are copied into the stream of the in-memory pages from the source host to the destination host. This means that the destination host is unaware which pages originate from the swap file and which pages come from in-memory (Figure 1.3). The host server thinks that the swapped pages are just memory pages that need to be stored and made available to the new virtual machine (Denneman, 2012).

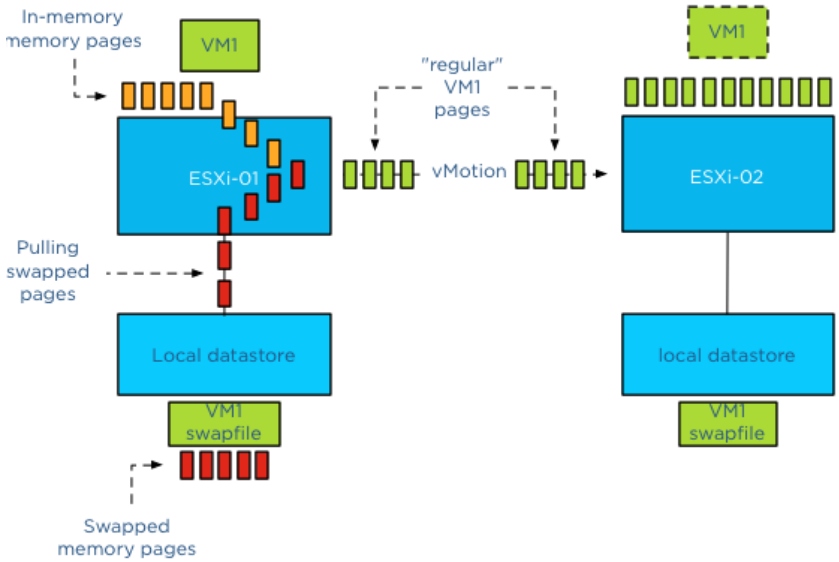


Figure 1.3: Swapped out pages are copying to the destination host (Deshpande et al. 2016)

According to Deshpande, the migration tool itself (such as QEMU in KVM/QEMU or xend in Xen) may need to compete with the VM's running applications for access to the swap device during the live VM migration process (Deshpande et al. 2016), and which may further increase the VM migration time and add strain on the data centre network.

1.2.2 Cloud Data centre Network Overhead and Competition for Bandwidth

Cloud computing has recently emerged as a highly successful alternative information technology due to its unique features such as on-demand resource provisioning, a pay-as-you-go business model, an unlimited amount of computing resources and high reliability. In order to meet high demand, cloud providers are deploying large-scale data centres compromising thousands of servers across the world. According to a recent report from the Cisco Systems, cloud data centres will dominate the global data centre traffic flow for the foreseeable future, they predict that more than four-fifths of the total data centre traffic will be cloud traffic by 2019 (Cisco,2015) .

One important fact that pointed out by the report is that a majority of the global cloud data centre traffic is generated due to the data communication within the data centres. A single data centre supports two types of traffic: traffic flowing between external systems and internal servers, and traffic flowing between internal servers (Greenberg, 2009).

A typical web application involves both traffic patterns at any given time, and using multiple applications in a cloud data centre generates a huge amount of network traffic. Another reason for generating traffic in a cloud based data centre is the intensive communication between VMs. When multiple applications are running on VMs, the communication needs to be synchronised with each virtual machine at regular intervals to maintain the consistent state of the VM (Kejiang Ye et al. 2012).

Generally Physical Machines (PM) in a large cloud data centre are connected to networks that have a higher bandwidth, but some applications which are running on VMs may experience huge user traffic during peak times. Migrating a VM from source to a destination PM generates a significant amount of traffic depending on the following factors; the VM's image size, its page dirty rates, the migration completion deadline and the available bandwidth. For example, let's assume that there are 20 VMs requiring migration within five minutes over 10 Gbps link, and if each VM consumes 1Gbps for 20 seconds, then it needs a 20 Gbps link to support over 400 seconds for the whole 20 VMs, which is clearly unacceptable. And what would happen if there was not enough bandwidth on the network to support live VM migration?

If VM migration occurs during those peak times, VM migration and user traffic will compete for network bandwidth, then the data centre's network may not have enough resources to support both VM migration and demands of application users, which would create a bottleneck in the network. Additionally an unscheduled large amount of VM migration may cause a data transfer bottleneck which ultimately impacts on the network overhead. Cloud data centre' users may have different QoS (quality of service) requirements; any unnecessary network bottleneck would cause performance degradation which may result in SLA (service level agreement) violations.

In this work, it is found that although there are techniques available for load balancing, VM scheduling, resources allocation, reduction of energy in data centres etc, but there is no specific automated framework for live VM migration to reduce network bottlenecks in cloud data centres by prioritising VM migration load and guaranteeing bandwidth for VM migration. Also a few attempts have made to dynamically map VMs to PMs and trigger VM migration automatically.

In addition, an uncontrolled VM migration degrades network performance as well as it wastes datacentres' resources. When migrating a VM form source to the destination host, the policy

adopted by the cloud service provider to allocate a VM to an available host is known as VM allocation policy. The problem of VM allocation can be defined as finding the optimal way to allocate a VM to a suitable host by maximizing the utilisation of resources of that host and minimising the overall wastage of resources at cloud datacentres. The proposed framework intends to increase the efficiency of VM allocation policy by allocating VMs to suitable hosts in an optimal way.

Given the analysis above, the main objective of this research is to present a solution for the problems that are highlighted in the problem statement. Therefore, this research presents a centralised, bandwidth aware, dynamic, intelligent and automated framework for Live VM migration. The framework is capable of balancing the load of the system through automated live VM migration, which manages the workload of physical servers as well as allocating the minimum bandwidth for VM migration by prioritising VM migration traffic over network user traffic. The detailed design of the proposed framework will be presented in the Chapter 5.

1.3 Research Objectives and Challenges

This thesis tries to identify and discuss challenges in relation to live VM migration in a cloud data centre, particularly the VM migration overhead, network overhead and bandwidth competition between VM migration load and the application's user traffic in a cloud based data centre. The goal of this thesis is to implement a network bandwidth-aware, intelligent, dynamic automated framework for live VM migration in order to reduce network bottlenecks on a cloud data centre's network by prioritising VM migration traffic over user traffic. This will in turn guarantee a minimum bandwidth for VM migration. In this thesis, particularly the following research problems are investigated:

- Detection of overloaded physical machines in a data centre. In a cloud data centre, it's very important to detect an overloaded host in order to provide reliability to users. In solving this problem, there is initially focus placed on the proposed framework and it is determined whether a host is overloaded or not.
- Detection of underloaded physical machines in a data centre. Detection of underloaded hosts in a data centre determines underutilised hosts. Underloaded machines can be used as the hosts for migrating VMs from overloaded servers.

- When to migrate VMs from overloaded hosts. This problem determines when to trigger VM migration; the crucial decision must be made to determine the best time to migrate VMs to avoid performance degradation.
- VM selection. Once an overloaded host is detected, it's important to move one or more of the VMs from the overloaded host to other servers in order to balance the load of the host. This problem determines the best VMs to migrate from overloaded hosts that will provide the most beneficial system configuration.
- Finding the best hosts for migrating VMs. This problem determines how to find the best host for migrating VMs.
- Calculating required bandwidth for migrating VMs. This problem determines how to find the minimum required bandwidth for migrating VMs.
- Calculating available bandwidth. This problem determines how to find available bandwidth on the network.
- How to prioritise VM migration traffic over user traffic? As it was discussed earlier, VM migration traffic and several different types of network traffic (user traffic, VM management traffic, shared storage traffic) on the data centre traffic would compete for the available bandwidth. This problem determines how to prioritise VM migration traffic over user traffic, and how it can allocate required bandwidth for VM migration if there is not enough bandwidth to support the VM migration.
- How to design following algorithms? The proposed framework's main concern is reducing the bottleneck on the data centre network, therefore this research takes the data centre network's bandwidth into consideration when designing novel algorithms. The framework consists of the proposed following algorithms.
    1. Overloaded host detection
    2. Underloaded host detection
    3. VM Selection
    4. VM Placement

To deal with the challenges that are associated with the above research problems, it is necessary to explore and analyse the research area of live VM migration with the following objectives which have been delineated.

- Conduct a competitive analysis in the research area of live VM migration to gain an understanding of the research issues
- Propose an approach to designing the proposed framework for live VM migration.

- Conduct a competitive analysis of existing overloaded host detection, underloaded host detection, VM selection and VM placement algorithms to obtain a theoretical insight into the designing of proposed algorithms.
- Installation of CloudSim simulator that can be used to evaluate the proposed algorithms.

1.4 Research Methodology

The research methodology followed in this thesis consists of consecutive steps summarised below:

- Carrying out a comprehensive literature review on live VM migration, and related work, then conducting a theoretical analysis on related work and existing algorithms.
- Since the proposed framework involves a set of algorithms, proposed algorithms are designed and developed based on the insight that gained from existing algorithms.
- Evaluate the proposed algorithms using the CloudSim simulator. As the Cloud computing environment which is intended to create a view of indefinite computing resources to users, it is important to evaluate the proposed algorithms on a large virtualised datacentre infrastructure. However, conducting repeatable large-scale experiments on a real infrastructure is not feasible. Therefore, to ensure the repeatability and predictability of experiments, and also to carry out large scale experiments, the CloudSim simulator has been selected as the initial method to evaluate the performance of the proposed algorithms.

- In this framework, the proposed algorithms are evaluated using following evaluation metrics:
  1. Number of overloaded hosts
  2. Number of migrations
  3. Number of underloaded hosts
  4. Number of SLA (Service Level Agreement) violations
  5. Energy Consumption

1.5 Contribution

The key contributions of this thesis can be broadly categorised in the following categories: analysis of live VM migration concept and using live VM migration as a load balancer, competitive analysis of existing load balancing implementations and algorithms, a proposed novel framework for live VM migration, and four novel algorithms for the proposed framework. The key contributions are:

- A survey of live VM migration concept.
- A survey of dynamic load balancing based on live VM migration, analysis of load balancing algorithms and VM load balancing implementations.
- A novel network bandwidth-aware intelligent dynamic automated framework for live virtual machine migration.
- Four novel algorithms, the main functions of the framework are based on these proposed four algorithms, which are:

1. Host overloaded detection algorithm
2. Host underloaded detection algorithm
3. VM selection algorithm
4. VM placement algorithm

- A novel SLA violation metric, bandwidth degradation due to VM migration (BDM)
- A conference paper, Virtual Machine Migration Strategy in Cloud Computing was published in 2015 (Liyanage et al.2015).

1.6 Thesis Organisation

This section presents the core chapters of this thesis, which are organised as follows. Chapter 2 presents a review and an analysis of Cloud Computing, and this chapter highlights enabling technologies behind cloud computing, types of cloud computing and deployments, key challenges and the future of cloud computing.

Chapter 3 presents a taxonomy and survey of live VM migration in cloud. This chapter highlights a live VM migration concept including techniques, VM memory migration, VM memory migration categories, performance metrics, live VM migration challenges and commercial implementations.

Chapter 4 presents dynamic load balancing based on live VM migration, an analysis of load balancing algorithms and VM load balancing implementations. This chapter highlights

different types of load balancing algorithms, their modelling and scheduling process. In addition this chapter discusses evaluation metrics and a depth analysis of existing load balancing implementations.

Chapter 5 presents the novel proposed network bandwidth-aware intelligent dynamic automated framework for VM live migration. It discusses the main components and their functions of the framework.

Chapter 6 presents the implementation and the analysis of the results. Here the four issues concerning proposed host overloaded detection, host underloaded detection, VM selection and VM placement algorithms are discussed. The proposed algorithms are implemented on a test bed in a CloudSim simulator. This section discusses the implementation process, evaluation metrics, simulation results and analysis. The chapter concludes with a summary of the main findings. Chapter 7 Presents the conclusion and indicates future work.

Chapter 2: Cloud Computing

This chapter starts by outlining the key technologies behind Cloud Computing and its evolution over the past decades. Then, it discusses the types of Cloud Computing services, most of which fall into to three broad categories: infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). This is followed by a comprehension discussion on Cloud deployments, because not all clouds are the same. There are three different ways to deploy cloud computing resources: public cloud, private cloud and hybrid cloud. As a new technology, Cloud Computing faces many challenges including technical, adaptation and other challenges. This chapter analyses cloud computing challenges that the Cloud research community must tackle in order to provide a secure and user friendly Cloud. Finally, the chapter discusses the future of cloud and it concludes with a critical analysis and summary.

2.1 Introduction

Today Cloud computing is emerging as a popular computing model which is transforming a large part of the IT industry. There is no standardised and uniform definition for cloud computing, Trieu et al defined cloud computing as a paradigm where computer processing, storage, and network capacity are made available to users in an on-demand manner through virtualisation on a shared physical infrastructure (Trieu et al.2010). The underlying theory of cloud computing dates back to the 1950s with gradual evolutions that started with mainframe computing. In the 1950s, a scientist by the name of Herb Grosch suggested that the entire world would operate on “dumb terminals powered by about 15 large data centres” (Irvin et al, 2012).

Developers or businesses with innovative ideas for internet services no longer require large capital outlays in computer hardware or software to deploy their services. Instead, they use the infrastructure and other services that are already provided and maintained for them by a cloud provider. Companies with large batch-oriented tasks can obtain a result as long as their programs can scale, since using 1000 servers for one-hour costs no more than using one server for 1000 hours. This elasticity of resources and without paying a premium for large scale of resources is unprecedented in the history of IT. As a result, cloud computing has become popular in recent years.

The main reason for the existence of different definitions of cloud computing is that it brings together a set of existing concepts to form a new business model and a set of services to meet demand of today's technology requirements. Cloud computing overlaps with many existing technologies such as Grid Computing, Utility Computing, Virtualisation, and Autonomic computing. The concept of cloud computing was first hinted at by John McCarthy in the 1960s, he stated that he believed, "computation may someday be organised as a public utility" (Garfinkel, 2011).

In the mid-1990s, the term Grid was used to describe technologies that attempted to implement several gigabit test beds to link super-computing sites across the United States. A computing grid is a distributed system that supports a virtual research environment across different institutions (Schwiegelshohn et al. 2010), and Grids are built to provide services to large scale scientific and business applications. The Grid computing model is shown in the figure 2.1 below.

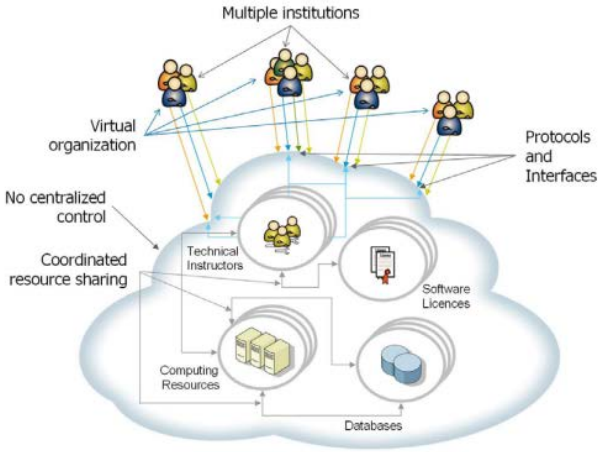


Figure 2.1: Grid Computing Model (Schwiegelshohn et al. 2010)

Autonomic Computing, originally coined by IBM in 2001, aims at building computing systems capable of self-management, i.e. reacting to internal and external observations without human intervention (Zhang et al. 2010). The idea of implementing autonomic computing was derived from the human body.

Utility computing represents the model of providing resources on demand and charging customers based on usage rather than a flat rate (Zhang et al, 2010). It enables customers to

buy computing capacity based on actual usage and demand. They can access the capacity according to their needs, whenever required and without expending resources or upgrading the capacity.

Examples of computational resources

- CPU usage time
- Physical memory and virtual memory
- Hard drive space and access time
- Network bandwidth

Cloud computing offers scalable and affordable compute utilities as on-demand service with variable pricing schemes. The pay-as-you-use economic model in Cloud is taken from the paradigm of Utility Computing. Users can use Cloud resources and pay according to the pay-as-you-use economic model; the users are bound by Service Level Agreements (SLA). SLA defines the contracted performance of the QoS (quality of service) and penalties for violation of the contracted QoS between user and the service provider (Huang et al.2016).

The main features of Cloud computing are the enormous amounts of power offered in terms of computing and storage while providing improved scalability and elasticity. Cloud computing distinguishes itself from other computing technologies in the following aspects (Shawish et al.2014).

- On demand service – Cloud computing offers resources and services for users on an on-demand basis. Users have the ability to customize and personalize their computing environments according to their requirements.
- QoS guaranteed offer - Service providers that provide computing resources and services can guarantee a QoS (Quality of Service) for users for example in CPU speed, memory size, storage size, bandwidth etc. The computing Cloud renders an ensured QoS by concluding a Service Level Agreement (SLA) with users.
- Autonomous Systems – Cloud computing technology is an autonomous system and it is managed transparently to users. In the Cloud, it has the ability to reconfigure hardware, software and other pools of resources to be presented as a single platform for users.

- Scalability – The scalability and flexibility are the most important features of cloud computing. It can be scaled across various aspects such as geographical locations, hardware performance, and software configurations.

There are different types of key enabling technologies behind cloud, but Virtualisation is the main technology behind cloud computing that supports simultaneous execution of diverse tasks over a shared hardware platform. Virtual Machine (VM) is a software implementation of a computing environment in which an operating system or program can be installed and run (Kapil et al. 2013). In a cloud computing environment, applications and services are hosted on Virtual Machines that span over several physical servers with dedicated resources (CPU cores, RAM, Disk Space, etc) are allocated to each VM in order to closely match the applications' needs. Virtualisation provides many benefits, such as resource utilisation, portability, application isolation, reliability, higher performance, improved manageability and fault tolerance (Kapil et al. 2013).

2.2 Enabling Technologies behind Cloud Computing Concept

Cloud Computing has been evolved by advancement in various technologies, A number of enabling technologies contribute to cloud computing, several state of art techniques are identified.

2.2.1 Virtualisation

The earliest use of VMs was by IBM in 1960, intended to leverage investments in expensive mainframe computers. The idea was to enable multitasking, running multiple applications and processes for different users simultaneously.

Virtualisation is the idea of partitioning or dividing the resources of a single server into multiple segregated VMs, and the resource virtualisation is at the heart of most cloud architectures (Antonopoulos et al. 2010). Virtualisation is the main technology that fosters the cloud computing concept and it enables multi-tenancy cloud business models by providing a scalable, shared resources platform for all tenants.

There are different types of virtualisation and each one is distinctive according to the element it is used on. As an example application Virtualisation provides a virtual implementation of the application programming interface (API) that is needed to modify and run applications, it

allows developers to run applications that are developed for one platform on another platform without needing to modify the application itself. However application virtualisation is outside the scope of this research.

2.2.1.1 Virtualisation Techniques

In general, there are three main virtualisation techniques, Full virtualisation, Paravirtualisation and Operating System Level virtualisation.

2.2.1.1.1 Full Virtualisation

In full virtualisation, the entire hardware environment is emulated by utilising hardware virtualisation support, binary code translation, or binary code rewriting, and also guest OS does not need to modify its kernel (Antonopoulos et al.2010). Having full virtualisation is important as a non-open source OS (Windows) can be run otherwise it is too difficult to modify non-open source kernels. Some of the current existing full virtualisation hypervisors are VMware, Xen, KVM, and Microsoft Hyper V (Palit et al.2013). Figure 2.2 shows the full virtualisation architecture

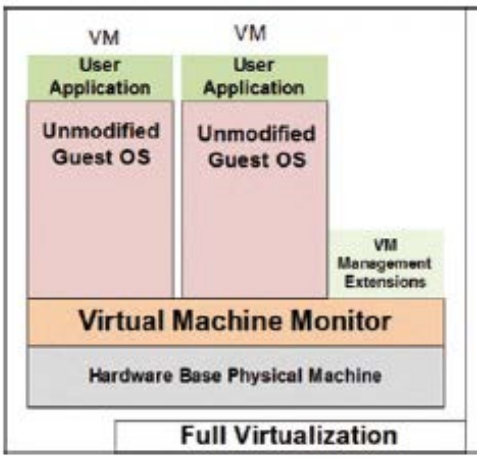


Figure 2.2: Full Virtualisation Architecture (Antonopoulos et al 2010)

In full virtualisation, one or more operating systems (OS) and the applications they contain are running on top of virtual hardware. Each instance of an OS and its applications runs in a separate VM (Virtual Machine) called a guest operating system (Scarfone et al, 2011). In full virtualization the hypervisor provides most of the same hardware as is provided by the hardware's physical platform.

Virtual Machine Monitor (VMM)

The OSs on a physical host are managed by the hypervisor which is also referred to as a virtual machine monitor (VMM). In virtualisation, the Virtual Machine Monitor (VMM) is a software layer that provides resources to emulate a hardware interface for the VMM to run on. The VMM runs on bare hardware or on top of an operating system, the hypervisor controls the flow of instructions between the guest OS and the physical hardware (Blenk et al, 2015). Figure 2.3 shows the VMM architecture.

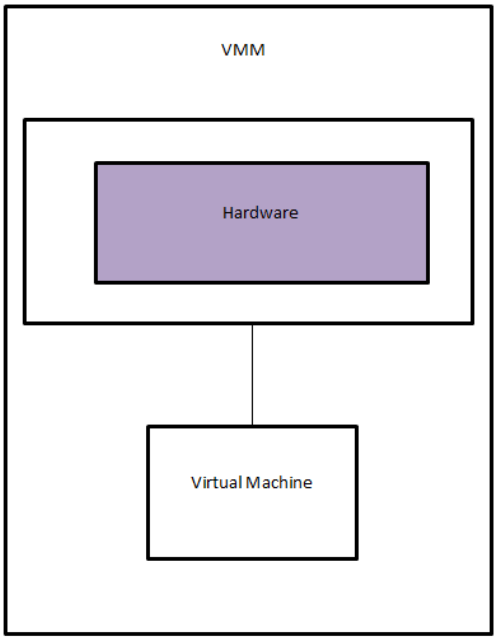


Figure 2.3: Virtual Machine Monitor (Hypervisor) Architecture (Kumar eta al. 2015)

Types of Virtual Machine Monitors

There are two types of full virtualisation, bare metal virtualisation (Type 1 hypervisor) and hosted virtualisation (Type 2 hypervisor). Type 1 hypervisors run directly on the machine's hardware with VM resources provided by the hypervisor. Type 2 hypervisors run on a host operating system to provide virtualisation services, in this research we only focus on type 1 hypervisors, e.g VMware ESXi and CitrixXenServer (virtualizationreview.com, 2009). Figure 2.4 shows the architecture of type 1 hypervisor (Bare Metal) and figure 2.5 shows the type 2 architecture.

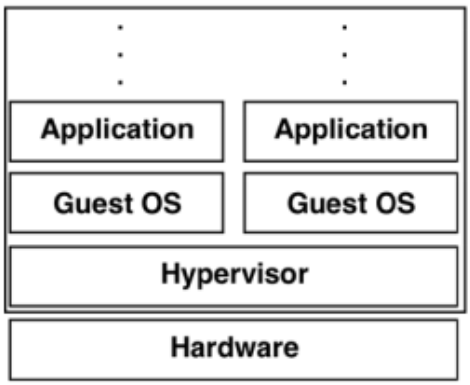


Figure 2.4: Bear Metal Virtualisation Architecture (type 1 hypervisor) (Scarfone et al.2011).

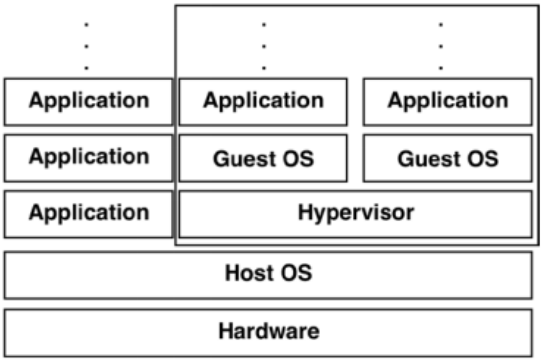


Figure 2.5: Hosted Virtualisation Architecture (type 2 hypervisor) (Scarfone et al.2011)

Servers are most often virtualised on physical machines using bare metal virtualisation (Type 1 hypervisor), Desktop are most often virtualised on computers using hosted virtualisation (Type 2 hypervisor).

2.2.1.1.2 Paravirtualisation

In Paravirtualisation, the running guest OS should be modified in order to be operated in the virtual environment (Sahoo et al. 2010). As paravirtualisation cannot support an unmodified operating system, its compatibility and portability are poor. The open source Xen project is an example for paravirtualisation that virtualises the processor and memory using a modified Linux kernel and virtualises the I/O using custom guest device drivers (VMware, 2007). Figure 2.6 shows the architecture of paravirtualisation.

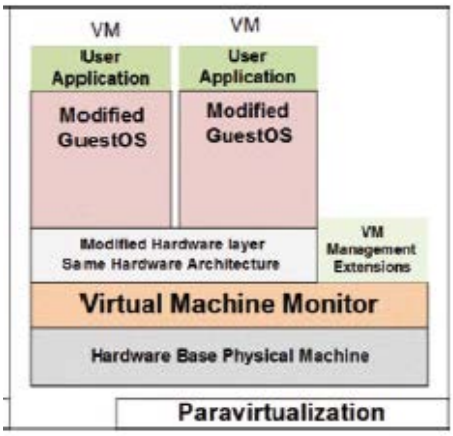


Figure 2.6: Paravirtualisation Architecture (Sahoo et al. 2010)

2.2.1.1.3 Operating System (OS) Level Virtualisation

It enables multiple isolated execution environments within a single operating system kernel. The main advantage of this approach is near native performance, density and features dynamic resource management. However, this approach requires guest VMs to share the same kernel as the host. On the other hand, this technology does not allow different type of kernels to be run at the same time. Some of the examples for OS level virtualisation are FreeBSD Jail, Solaris Zones/Containers, Linux-VServer, Open VZ and Virtouzzo (Kolyshkin, 2006).

2.2.1.2 Types of Virtualisation

The term "virtualisation" has become ubiquitous representing any type of process of obfuscation where a process is somehow removed from its physical operating environment. Because of this ambiguity, virtualisation can almost be applied to any and all parts of an IT infrastructure.

2.2.1.2.1 Operating System (OS) Virtualisation

This is the most common form of virtualisation; the hypervisor encapsulates all of the components of a guest Operating System, including its application and the virtual resources they use, into a single logical entity (Scarfone et al.2011). This empowers users to reduce the amount of physical hardware required to run their applications and software by cutting down the number of physical machines.

2.2.1.2.2 Virtualised Networking

In recent years, the concept of network virtualisation attracted significant attention; full virtualisation hypervisors can provide networking capabilities. Typical hypervisors provide three primary forms of network access.

- Network Bridging
- Network Address Translation (NAT)
- Host Only Networking

When a number of guest OS are running on a single physical host, the hypervisor can provide a virtual network for these guest OS by implementing virtual switches, hubs, and other network devices. Virtualised networking provides a networking environment that allows multiple service providers to dynamically compose multiple heterogeneous virtual networks that coexist together in isolation from each other (Chowdhury et al. 2009). Some of the virtualised networking technologies are described below.

- Virtual Local Area Networks (VLAN)
- Virtual Private Networks(VPN)
- Layer 1,2, and 3 VPN
- Software Defined Networks

Network virtualisation has the same goals when it comes to the network fabric that connects virtual servers. Network virtualisation should allow a virtual network, including all of its IP addresses, routers and network appliances to appear to be running directly on the physical network (Benmessaoud et al. 2014). Network virtualisation allows the servers that are connected to the virtual network to operate as if they were running directly on the physical network even though multiple virtual networks share the physical network. According to Benmessaoud et al. the key benefits of network virtualisation are stated below (Benmessaoud et al.2014).

- Ability to run multiple virtual networks securely, and isolated from each other all with the illusion that they are each alone on the physical network.
- The ability to move Virtual Machines (VMs) around in the physical network without having to reconfigure the physical network, including IP address and VLAN's.
- The ability to abstract the virtual network away from the underlying physical network.

2.1.1.2.3 Storage Virtualisation

Hypervisors have many ways of simulating disk storage for guest OSs. All hypervisors, as a minimum have virtual hard drives while some of them have more advanced virtual storage options. Today, several techniques are employed to virtualise different storage functions within one model, some hypervisors can use advanced storage interfaces on the host system, such as network attached storage (NAS) and storage area networks (SAN). In general, there are two types of storage technologies, Block-Level and File-Level and storage virtualisation can be structured in three methods: host based, storage device based, and network based.

In Block-level storage, servers are using an industry standard Fibre channel and iSCSI connectivity mechanism. In this technology, raw storage volumes are created, and then the server-based operating system connects to these volumes and uses them as individual hard drives (Lowe, 2011). Flexibility and versatility of block-level storage make it usable for any kind of application including file storage, database storage, virtual machine file system (VMFS) volumes and more. Also it can place any kind of file system on block-level storage, for example if it's running Windows, volumes will be formatted with NTFS; and VMware servers will use VMFS.

File level storage virtualisation refers to provisioning storage volumes to operating systems or applications in form of files and directories. File level storage is usually accessible using common network protocols such as SMB/CIFS (Windows), NFS (Linux, VMware), Common Internet File System (CIFS) and Network File Systems (NFS). File level storage virtualisation is a file presentation in a single global namespace (Tate.J et al. 2016).

Host based approach relies on an agent or management software installed on one or more host systems. Physical drivers are handled by a traditional device driver while a software layer above the device driver handles I/O (input output) requests, looks up metadata and redirects I/O.

Storage device based – in this setup, virtualisation can be built into the storage fabric: an example, newer RAID controllers allow other storage devices to be attached downstream.

Network Based – In this configuration, storage virtualisation is viewed as a network based device, generally using Fibre channels networks connected to a SAN (Storage Area Networks) Generally, a fibre channel switch is placed between the host and the storage, and this switch virtualises and redirects all IO requests (Kay, 2008).

2.1.1.2.4 Desktop Virtualisation

Desktop virtualisation sometimes referred to as Client Virtualisation, which is defined as virtualisation technology that is used to separate a computer desktop environment from the physical computer. Desktop virtualisation is a type of client/server computing model, because the virtualised desktop is stored on a centralised and remote server and not on a physical machine that is virtualised (Tate et al. 2016). A virtualised desktop can be accessed remotely from any location and also users can interact with a virtual desktop in the same way that they access and use a physical desktop.

2.2.2 Service Oriented Architecture (SOA) and Web Services

Competiveness of the IT industry requires that companies continually modify their IT system by adding new features of removing legacy software or applications in a relatively short period of time. Traditional software lifecycle models haven't explicitly addressed this requirement for continuous integration of new capabilities. Service Oriented Architecture allows companies to construct rapid, low-cost, secure, reliable software and applications. It reduces the need to develop new software components each time a new business process arises.

Service Oriented Architecture (SOA) is defined in many literatures with extensive number of articles attempting to define what it means and how it can be used and implemented in an organisation. SOA is a collection of independent loosely coupled applications that are capable to communicate in the form of provision of service (Josuttis, 2007). It is a collection of services and these services communicate with each other, the communication can involve either simple data passing or it could involve two or more services coordinating some activity. Service Oriented Architecture provides an approach for building distributed systems that deliver application functionality as services to either end user applications or other services. When SOA principles combined with Cloud computing, it offers full service driven infrastructure for many applications.

Services

If a SOA is to be effective, we need a clear understanding of the term of service. A service is software and hardware and it can be used in multiple ways, there are two types of services: atomic and composite. An atomic service is a well-defined, self-contained function that not depends on the context or state of other services. A composite service an assembly of atomic or other composite services, a service within a composite service may depend on the context or state of another service that is also within the same composite service (Barry et al. 2013). Figure 2.7 shows the architectural stack and the elements that might be observed in service oriented architecture.

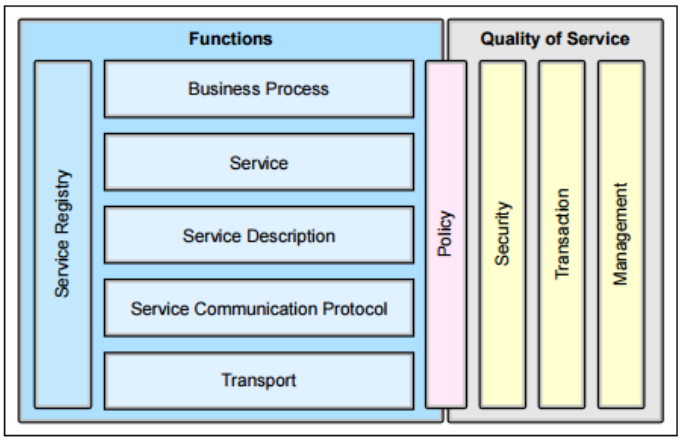


Figure 2.7: Elements of a Service Oriented Architecture (Endrei et al. 2004)

The architectural stack is divided into two parts: Functions and Quality of Services and these elements are described in detail below (Endrei et al. 2004).

Transport – Is the mechanism that used to move service request from the service consumer to the service provider and the service responses from the service provider to the service consumer.

Service Communication Protocol – Is an agreed mechanism that the customer and the service provider use to communicate what is being requested and what is being returned.

Service Description - This is an agreed schema for describing what the service is, how it should be invoked, and what data is required to invoke.

Service – A service is an actual service that is made available for customers to use.

Business Process – Is a selection of services, it has particular set of rules and invoked in a particular sequence to meet a business requirements.

Service Registry - Is a repository service which may be used by service providers to publish their services.

Quality of service includes:

Policy – Is a set of conditions and rules under which a service provider makes the service available for customers.

Security – Is the set of rules that might be applied to the identification.

Transaction – Is the set of attributes that might be applied to a group of services to deliver a consistent result.

Management – Is the set of attributes that might be applied to managing services provided or consumed.

Web Services

The term web services has various, imprecise, and evolving meanings. "A web service is a software application identified by a URI (Uniform Resource Identifier), whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A web service supports direct interactions with other software agents using XML based messages exchanged via internet based protocols (Endrei et al. 2004)."

As the name suggests, a web service is a kind of webified application that typically delivered over HTTP (Hypertext Transport Protocol) or HTTPS (Hypertext Transport Protocol Secure) (Kalin, 2013). Web services can be programmed in a variety of languages, old and new. In more technical terms, a web service is distributed software system whose components can be deployed and executed on physical distinct devices. As an example, a web server that hosts a web service and a mobile device client that hosts an application request a service from the web server (Figure 2.8).

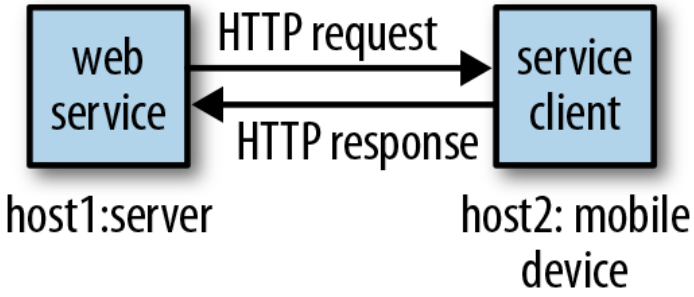


Figure 2.8: A Web Service and one of its Clients (Kalin, 2013)

Web services come in two popular flavours: SOAP (Simple Object Access Protocol) based and REST (Representation State Transfer) style.

SOAP (Simple Object Access Protocol)

SOAP was originally part of the specification that included Web Service Description Language (WSDL) and Universal Description, Discovery, and Integration (UDDI), a combination of WSDL, UDDI and SOAP formed the original Web Service specification. SOAP is now used without the UDDI and WSDL, instead of the discovery process, SOAP messages are hard-coded or generated without the use of a repository.

SOAP provides the envelope for sending Web services messages over the internet, and it's part of the set of standards specified by W3C (Barry, 2007).

The SOAP envelope contains two parts (shown in the figure 2.9).

- Header – Provides information on authentication, encoding data or how a recipient of a SOAP message should process it.
- Body – Contains the message

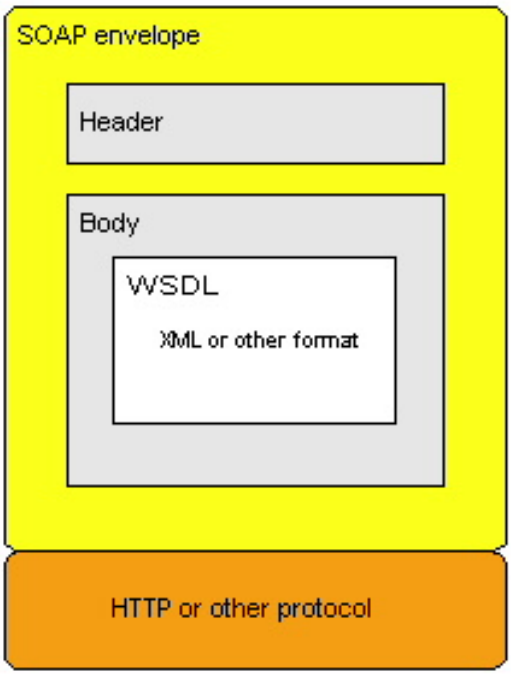


Figure 2.9: SOAP Envelope (Barry, 2017)

REST (Representation State Transfer)

REST is a style of architecture based on a set of principles that describe how networked resources are defined and addressed (Barry, 2007) .REST applications and architectures are sometime referred as RESTful or REST-style applications or architectures. REST has proved to be a popular choice for implementing Web Services. REST appeals to developers because it has a simpler style that makes it easier to use than SOAP. The operation of REST web services is shown in the figure 2.10.

Figure 2.10: Using REST for Web Services (Barry, 2017)

A service oriented architectural approach is the evolution of a system or software architecture for addressing componentisation, reusability, extensibility, and flexibility. In order to construct scalable Cloud Computing platforms, it is necessary to leverage SOA to build reusable components, standard based interfaces, and extensible solution architecture (Zhang, 2012). SOA architecture is important to cloud computing for a few key reasons.

SOA is a good architectural approach that deals with the proper formation of information systems. SOA binds with cloud computing to deliver cloud based services, and cloud computing relies on service-oriented architecture. In order to take advantage of cloud computing, interfaces and architectures are needed that can reach out and touch cloud computing resources. SOA architecture provides sets of services and these services communicate with each other, the communication can involve either simple data passing or it could involve two or more services coordinating some activity. Another important point is that enterprises or organisations need some sort of architectural discipline with guiding principles to document and organise their architecture. And SOA is a good approach if it follows the correct steps.

Both cloud computing and SOA share concept of service orientation with different types of services which are available thorough cloud computing. Cloud computing focuses on turning aspects of the IT computing stack into commodities that can be purchased incrementally from cloud based providers. On the other hand, SOA is not restricted conceptually to software, it is often implemented in practice as component or software services. As an example, Web Service is a SOA implementation and Web Service standards are used in many cloud implementations.

A combination of SOA and cloud bring benefits to enterprises, as an example, a leading UK food retailer, a FTSE 500 company used a bottom-up SOA approach with a cloud based integration solution to bring existing customer and product data into the company's new Salesforce solution. As a result, the company has a single view of the customer support multi-channel marketing online, in the call centre and at the supermarket checkout register (Mulesoft.com, 2013).

Cloud computing and SOA work hand in hand to ensure success in the enterprise, cloud computing brings a level of maturity through its cloud services and operational foundation. On the other hand, SOA lends the experience of being a good enterprise citizen through governance and interoperability.

2.3 Types of Cloud Computing Services

According to the business model adopted, cloud is usually classified into three major categories, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Antonopoulos et al.2010). Those services are made available as subscription-based services in a pay-as-you go model to consumers.

The cloud computing architecture can be divided into four main categories: Hardware layer, Infrastructure layer, Platform Layer and Application layer (Zang et al. 2010), figure 2.11 illustrates the cloud computing architecture.

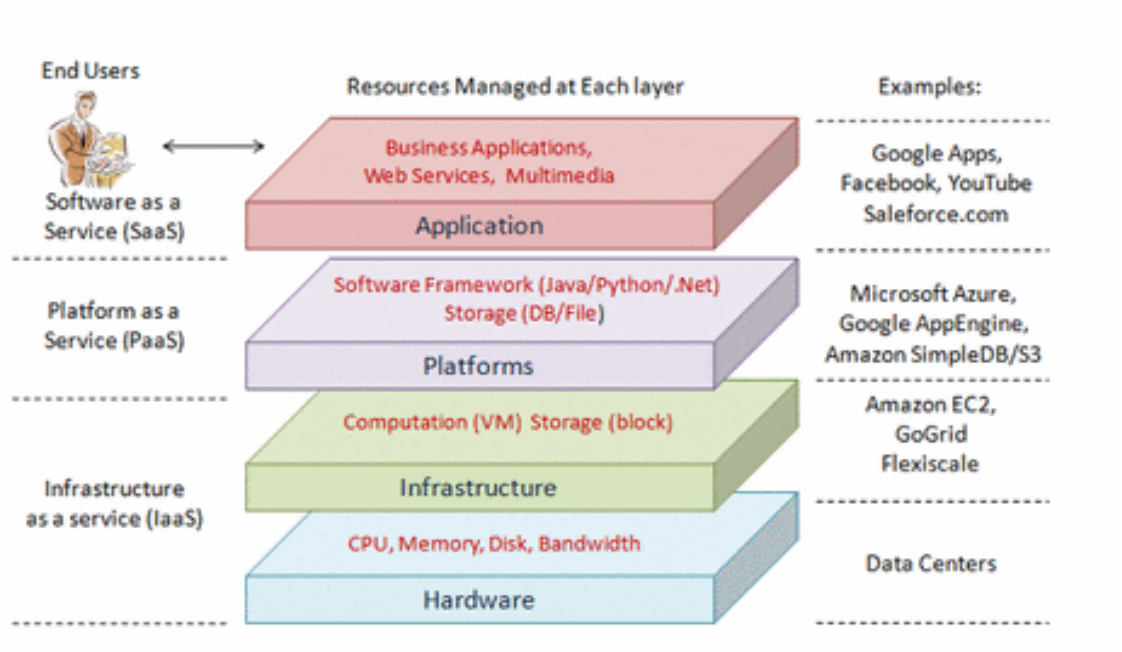


Figure 2.11: Cloud Computing Architecture (Zhang et al. 2010)

Hardware Layer - This layer is responsible for maintaining physical resources of the cloud; it includes physical servers (PM), routers, switches, and power and cooling systems. Infrastructure Layer - This layer is also known as the virtualisation layer, it facilitates resources to users by partitioning the physical resources. Platform Layer – This layer provides an environment for cloud users to run their software, as an example, users can run online applications such as Google App Engine (GAE) and they can use this layer as a development platform for their applications and software. Application Layer – This layer is

also called Software as a Service (SaaS), it refers to providing on-demand applications and software to users over the internet, examples, MS Office 365, Salesforce.com.

2.2.1 Software as a Service (SaaS)

SaaS is the top layer provider in which customer with ready to use applications running on the infrastructure provider. SaaS can be explained as a process by which the Application Service Provider (ASP) provides different software applications over the internet (Dhir, 2017). Users can utilise a web browser to access software that resides, along with the programs and user data in the cloud. Companies that use SaaS solutions eliminate the need for in-house (data-center-based) applications, administrative support for applications, and data storage. Because SaaS solutions reside within the cloud and the SaaS solutions can easily scale to meet customer needs. It enables the customers to bypass installing and operating the application on their own computers and buying the licence, and it also gets rid of the immense load of software maintenance. The SaaS provider provides a licence to the user as a service on demand through subscription. Generally the user is only able to modify parameters of the application that have been exposed by the provider.

SaaS applications are often multitenant solutions; that is, within the cloud, two or more companies may share the same sever resources (Jasma, 2013) as shown in the figure 2.12. Multi-tenant architecture (MTA) allows multiple customers (i.e tenants) to be considered into the same operational system where users run and share the same application instance as well as costs, which are significantly reduced.

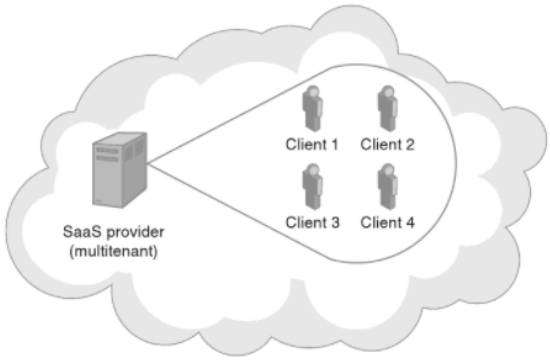


Figure 2.12: SaaS Multitenant Environment (Jasma, 2013)

There are four main SaaS architectures, multi-tenant architecture SaaS can be implemented via different ways, which are (WeiTek et al. 2014) stated below.

- Database-oriented SaaS - As the name suggests, this approach provides database-based SaaS solutions. It contains database-based and metadata-driven architecture to implement MTA.
- Middleware-oriented architecture - SaaS middleware-based approach whereby an application request is sent to a middleware that passes the request to databases behind the middleware. As all databases are behind the middleware, all applications requests to databases are managed by the middleware.
- PaaS based architecture - SaaS developers use an existing PaaS such as Microsoft Azure or Google App Engine (GAE) to develop SaaS applications. Many SaaS vendors provide their application platforms (PaaS) for SaaS applications, for an example Force.com provides a set of pre-defined data objects, such as customer, orders and accounts. But these are available for the CRM and related domains only.
- Service-oriented architecture - One can use a service-oriented approach to implement MTA where SaaS infrastructure can be developed using service components.

One of the popular SaaS Examples is Salesforce.com which is an example of SaaS database-oriented architecture, a representative system of the middleware approach is Corenttech.com. Google App Engine is an example for PaaS based SaaS, Microsoft Office 365, and Dropbox are some examples of SaaS.

2.2.2 Platform as a Service (PaaS)

Platform as a service provides developers with a platform, including all the system and environments, compromising the end-to-end lifecycle of developing, testing, deploying, and hosting of web applications (Antonopoulos et al. 2010). It provides an easy way to develop business applications and various services over the internet. Application developers don't need to maintain their own infrastructure, instead they can use different platforms that are already implemented by cloud providers and available for users. Creating and maintaining an infrastructure is the most time consuming and expensive work for developers. PaaS is especially useful for situation where multiple developers working on a development project. In PaaS, the user doesn't control the underlying cloud infrastructure including network, servers, operating systems, or storage, but it controls over deployed applications and possibly

configuration settings for the application-hosting environment. Some of the examples for PaaS are, Microsoft Azure. Google App Engine (GAE), Heroku.com, Salesforce.com, Force.com, Rackspace sites, Bungee Connect. (Dhir et al. 2017).

Benefits of PaaS

- Reduce the time that taken to develop application
- Fast deployment of applications to the market
- Integration with web services and databases via common standards
- Multiple users from various locations can work together
- Makes development possible for “non-experts”.

2.2.3 Infrastructure as a Service (IaaS)

Infrastructure as a Service is the delivery of computer infrastructure as a service for users over the internet. IaaS can be utilised by customers to create cost affective and easily scalable IT solutions where the complexities and expenses of managing the underlying hardware and outsourced to the cloud provider (dhir et al. 2017). IaaS provides machines, storage, network resources that developers can manage by installing their own operating systems, applications and support resources. Figure 2.13 illustrates an IaaS model.

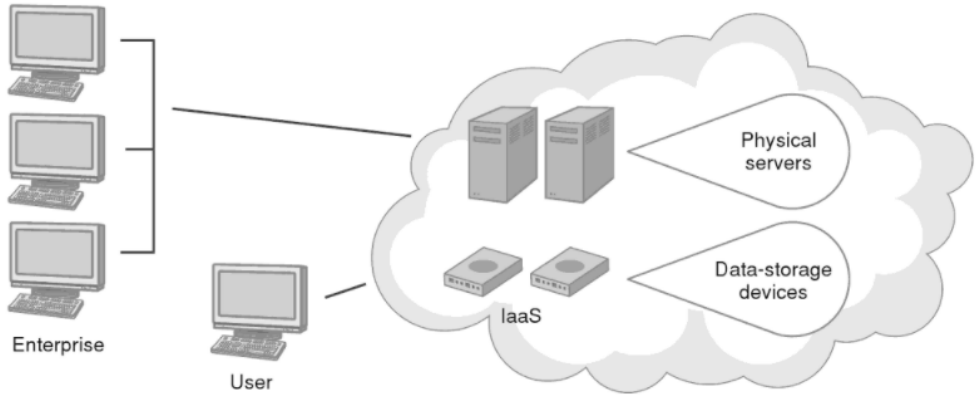


Figure 2.13: IaaS Model (Jasma, 2013)

A key benefit of IaaS is the usage-based payment scheme, which allows users of IaaS to pay as they utilise the resources. Another advantage is that of always using the latest technology and customers can achieve a much faster service delivery and time to market. Some of the examples for IaaS are Microsoft Azure, Amazon EC2 and Rackspace (Jasma, 2013)

2.3 Types of Cloud Deployment

According to NIST (National Institute of Standards and Technology), there are four cloud deployment models with different characteristics to serve different type of customer and their requirements, the models are the following,

- Public Cloud
- Private Cloud
- Hybrid Cloud
- Community Cloud

Public Cloud

The NIST definition for public cloud is: "*The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organisation or some combination of them. It exists on the premises of the cloud provider and is a form of providing public cloud services and a cloud service provider's business model brings economy of scale in pooling datacentre resources, virtualisation and on-demand provisioning allows outsourcing enterprise IT infrastructure addresses disaster recovery problem suitable for SME and agile companies*" (Simpson, 2017).

These clouds are managed by a third party organisation that guarantees the Quality of Service (QoS) as long as the users have enough bandwidth to cope with cloud traffic. In terms of the architecture, there is not much difference between a private cloud and a public cloud. But a public cloud can be accessible by anyone who wants to use it, usually software and services on a public cloud are offered on a "pay as you go" basis. A public cloud has a pool of virtual and physical resources that can be accessed by consumers on demand but users don't have control over the management of public cloud resources. Figure 2.14 shows the public cloud model.

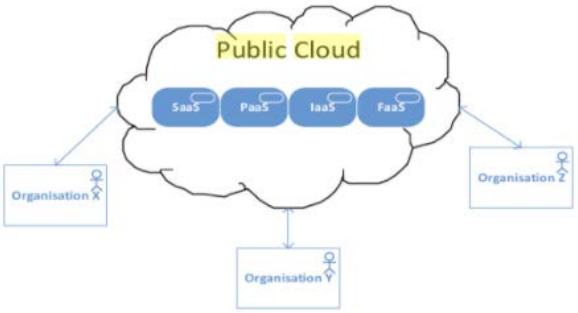


Figure 2.14: Public Cloud Model

Private Cloud

As the name suggests, a private cloud is used solely by a single organisation, but a private cloud consumer organisation or provider or a combination of both can establish, operate, manage and support the private cloud. It's a dedicated environment that can be hosted either on-site at the consumer's premises or at a service provider's data centre (Gill, 2015). Computing services can be accessed over the internet or a private internal network and to selected users only instead of the general public. The private cloud also referred as an internal or corporate cloud, it seems appropriate for large organisations that need higher level of security and privacy through both company firewalls and internal hosting to ensure that operations and sensitive data are not accessible to third-party providers. There are two cloud services that can be delivered in a private cloud, the first is Infrastructure as a Service (IaaS) that allows a company to use infrastructure resources such as compute, network and storage as service. The second is, Platform as a Service (PaaS) that lets a company deliver everything from simple cloud-based applications to sophisticated enterprise applications.

Hybrid Cloud

A hybrid cloud is a combination of a private cloud and a public cloud to perform distinct functions within the same organisation (Gill, 2015). Public cloud services are more likely to be cost effective and scalable than private clouds, therefore an organisation can maximise their efficiencies by employing public cloud services for all non-sensitive operations. A private cloud can be used for managing sensitive information, which offers higher level of security and privacy.

Amazon Web Services (AWS) rolled out a cloud service called Virtual Private Cloud (VPC) which is a secure and seamless bridge between organisation's existing IT infrastructure and

the Amazon public cloud. It is positioned as a mixture between private cloud and public cloud (Dillon et al. 2010).

Community Cloud

Community cloud is provisioned for a specific community that has mutual goals and requirements, and those organisations jointly construct and share the same cloud infrastructure as well as policies, requirements, values, and concerns (Dillion et al. 2013). The cloud infrastructure could be hosted by a third party cloud provider or one of the organisations in the community. A community cloud can implemented as a private or public community cloud. A typical private community cloud can be implemented for government agencies. Several government agencies may provision a private community cloud to share computing resources, cost and benefit. It can be managed by the lead government agency.

2.4 Adaptation of Cloud Computing

As cloud computing continues to improve, offering different services, the enterprise is steadily moving toward the cloud IT by incorporating all the cloud concepts. Having understood the significance of cloud embarkation, enterprises are busy in cloud assessment, enablement and on-boarding activities. Cloud computing has brought in innumerable tectonic and trendsetting shifts for both IT and business. Though it is an evolutionary idea and a new computing paradigm, it becomes more popular among enterprises, because it offers proven, potential and a promising new array of enterprise technologies (Raj, 2012).

The cloud computing has greatly impacted every enterprise these days. Enterprises are fast strategizing to absorb all the advantages and opportunities that the cloud offers to business. The closer and tighter alignment and association between business and cloud will take any cloud enterprise to greater heights.

According to Giannakouris et al, it stated that 97% of enterprises used cloud computing in 2016 (Giannakouris, 2016). Of the enterprises that reported using cloud computing, some 65 % relied on a cloud solution for their email. Instead of setting up their own email server and the infrastructure, most companies opted for cloud based solutions on a per-user operating cost. 62 % of enterprises used to store their files while 44 % used cloud database applications. Most importantly, more enterprises accessed advanced cloud applications such as CRM (customer relationship management). Compared to 2014, the increase in the use of cloud

computing was highest in the IT field (+ 9 %) while in the professional, scientific and technical sector +7%, followed by the hospitality sector + 5 %.

According to a survey conducted by Vanson Bourne in 2011, researchers asked 450 senior managers who are responsible for making IT purchasing decisions what they thought about cloud services. The senior managers were from broad spectrum of organisations from both the public and the private sector. It found that 48 % of organisations use some sort of cloud based solutions, with the greatest use coming from private companies with more than 200 employees (Bourne, 2011).

Cloud computing appears to be taking off fast in the UK, and while data suggests the vast majority of enterprise customers are becoming increasingly comfortable with cloud-based solutions According to a survey which was conducted in February, 2017 by Cloud Industry Forum, researchers polled 250 IT and business decision makers in large enterprises including small to medium sized business (SME) and also the public sector. The result revealed that the overall cloud adoption rate in the UK now stands at 88%, with 67% of users expecting to increase their adoption of cloud services over the coming year (Ismail, 2017). The majority of respondents (58%) described their organisations as using Hybrid cloud. These researches highlighted the extent of change in the IT landscape during last six years in UK, and also how organisations in the private sector are increasingly and consistently warming to the cloud model.

Six or seven years ago, the term “Cloud” didn’t even exist in UK government circles. In the government sector, IT services were typically provided by a small number of large IT contractors based on long term contracts. In the UK public sector IT has never been lower. According to a report that published in July, 2011 by Public Administration Select Committee (PASC), the government’s whole processes for procuring IT projects were fatally flawed. It listed several failed big IT projects that have run late, under-performed, or failed over the last 20 years. The following are some of the failed projects in the government’s IT sector during the last 20 years.

- The Child Support Agency’s IT System.
- National ID card Scheme.
- The Defence Information Infrastructure Programme.
- The Single Payment Scheme by the Rural Payment Agency.
- The National Offender Management System.

- NHS National Programme for IT.

How can this cycle of failure be broken? With the ability to drive efficiencies, improve flexibility and scalability and reduce cost, cloud computing offers a great deal to the UK's increasingly pressurised public sector. The government has already declared a willingness to adopt cloud computing in some form. According to a white paper released by the Cloud Industry Forum in 2015, around four in five public sector organisations formally adopted at least one Cloud service (78%) in the UK, up from 74% a year previous and a rise from 38 % in 2010 (Outsourcery.co.uk,2015). Looking to the future ahead, the public sector's engagement with Cloud Services looks set to increase significantly.

The level of cloud adaptation for organisations in developing countries looks very different from those in developed countries. According the Information Economy Report that was published by United Nations Conference on Trade and Development (UNCTD) in 2013, the market for the cloud computing in developing countries is expanding and the revenue is considerably larger. The UN estimated that it has potential implications for both the supply and the user side of the cloud economy (UNCTD.org, 2013).

In developing countries, there has been extensive adaptation by individuals for cloud services such as webmail and social networks, but for several reasons, cloud computing adaptation in the developing world has been inhibited. The main cloud adaptation challenges in developing countries are poor infrastructure and lack of access to bandwidth. Average fixed broadband connection is more than 28 subscriptions per 100 people in developed economies, 6 in developing countries and only 0.2 in the least developed countries (UNCTD.org, 2013).

The cost of communication remains another critical factor for adaptation of cloud services. The cost of Internet, cloud services, hardware and software are likely to form a much higher proportion of the total cost of cloud provisioning than in advanced economies.

2.5 Key Challenges in Cloud Computing

Cloud computing has generated significant interest in both the academic world and industry, but it is still a novel and evolving paradigm. Nevertheless, enterprises are moving their assets to the cloud to capture its business benefits, however, cloud computing presents some pitfalls with a number of challenges and concerns.

2.5.1 Security and Privacy

The top most concern and challenge in cloud computing is cloud security. According to the Cloud Security Alliance (CSA), over 70 % of the world's business now operate at least some parts of their businesses in cloud computing (Ma, 2015), but many businesses are concerned about the security, reliability, availability, and the control over their own data.

Sharing the cloud with other users possesses risks and concern over cloud security. As an example, business customers are concerned how their data is segregated if their data is reside on the same physical machine that shared by other businesses. Would data stealing spread from one tenant to another? Or from one virtual machine to another? How is customer data privacy preserved in this environment?

Security overall covers three aspects: Confidentiality, Integrity, and Availability (CIA) (Vyas, 2017). These aspects are the topmost consideration when designing cloud security.

- Confidentiality: Protecting data and information from disclosure to unauthorised persons.
- Integrity: Protecting data and information from being modified by unauthorised persons.
- Availability: Authorised users are able to access and use data whenever they require.

Some of the top security concerns in cloud computing are;

Data Breaches

This one of the most common security threats in IT security, a study conducted by the Ponemon Institute reported that over 50 % of the IT and security professionals surveyed believed their businesses security measures to protect data on cloud service are low. Those studies used nine scenarios, and after evaluating nine scenarios, the report concluded that overall data breaching was three times more likely to occur for businesses that utilize the cloud (Ma, 2015). The simple observation is that the cloud comes with unique set of characteristics that it makes it more vulnerable.

Hijacking Accounts

It is generally a practice to steal and highjack a person's cloud account information associate with a cloud service (Kumar et al. 2017). In April 2010, Amazon faced an across-site scripting bug that targeted customer credentials (Ma, 2015).

Malicious Insiders

The attacker can be anyone that related to the user or a friend, or an employee within the organisation that can retrieve one's personal information with the intent of damaging or exploiting the user's credentials (Kumar et al.2017).

On the 4th of February, 2015, a health insurance company Anthem Inc in the US disclosed what was by far the largest data breach in healthcare history. The cyberattack involved hackers who stole personal information including names, birth dates, and social security numbers of 78.8 million of its customers and employees (Herman, 2016). Investigators found that hackers had smuggled out data from a cloud-based file sharing service.

Code Spaces is not a well-known company, and its data breach did not affect millions of people but it was a good example of a company put completely out of business by a single cloud security incident. Code Spaces was a small company, a sub division of a Git (open source vision control system) hosting provider and used by project management and development organisations. By the time the incident occurred, it was making a name for itself as an IaaS provider. It started as a DDoS (denial of service attack) and when Code Spaces reached the attacker, they were told to pay a ransom. When the company declined, the hackers started destroying their resources until there were barely any remaining. According to the report, Code Spaces stated that "we finally managed to get our panel access back but not before he had removed all EBS snapshots. In summary, most our data, backups, machine configurations, and offsite backups were either partially or completely deleted" (Ragan, 2014).

Apart from those two stories and other few isolated incidents, it is worth knowing how many security breaches didn't involve the cloud. In fact many of the disastrous data breaches didn't involve the cloud at all. At the beginning of 2010, the cloud was regarded as too risky for medium to large size organisations to use, but cloud security has been improved since, because the best cloud providers are investing a much larger sum of money and resources on cloud security than the average company can.

2.5.1.2 Lack of SLAs (Service Level Agreement) for Cloud Security

The cloud offers many diverse options for users on the IaaS and PaaS layers, but cloud users face a daunting task when trying to select cloud resources that meet their requirements and QoS (quality of service). QoS guarantees the level of performance (e g bandwidth, I/O operations, storage) reliability, cost, security and availability offered by cloud providers, (Pothuri et al. 2016). QoS is fundamental for cloud users who expect cloud providers to deliver the advertised quality characteristics. Then cloud providers and cloud customers have to negotiate a Service Level Agreement (SLA) that allows them to formally specify the QoS requirement. SLA is an agreement between the cloud provider and the customer. SLA denotes a service commitment whereby the cloud provider strives to meet a service commitment specified over a time period (Murugesan et al. 2016). The service commitments include availability (e.g, 99.9 % of a VM), response time (less than 50 micro seconds for a storage request), disaster recovery (e.g. within 24 hours), ticket resolution (e.g within 1 hour of reporting), and data backup (e.g every day), naming some of the examples. Failure to achieve those metrics will result in a service credit to customer, service credit is given to the customer if one or more service guarantees are not met (Murugesan et al. 2016). These credits can be full or partial credit to a cloud customer resulting from service disruption. For a cloud customer, detecting and measuring SLA violations requires storage of request logs, service request metadata, and any other relevant data.

However, security terms are not covered in SLAs of the public cloud providers, especially security aspects such as confidentiality and integrity (Carvalho et al. 2017). For example, the Amazon EC2 SLA only specifies service availability, without any other QoS assurance. It's reasonable to expect that not all providers will be able, or willing to provide the same level of security for their customers, but SLA could be extended to cover security aspects of cloud computing with a defined security level.

2.5.2 Data Governance

According to the Data Governance Institute (DGI), data governance definition is " *Data governance is a system of decision rights and accountabilities for information-related processes, executed according to agreed-upon models which describe who can take what*

actions with what information , and when, under what circumstances, using what methods" (cloudindustryforum, 2016) .

The trust is the heart of the problem for many businesses when it comes to moving to cloud. Some businesses are worried that cloud providers will not bring industry best practices to the table. When organisations move their data to cloud, it's vulnerable to disclosure or loss. Some businesses will not know where their data is being held. Some countries regulations prevent data being sent across national boundaries, as an example, in Germany, passing data across national boundaries can be a federal offence. Therefore, all cloud users need to have an idea of national laws and regulations, and cloud providers must bound to disclose the place where users data being stored. But some low cost providers may try to blur the issue of where data is being held by using content delivery networks (cloudindustryforum, 2016) or wide area data accelerators.

2.5.3 Interoperability Issues

Interoperability means easy data migration and integration of application and data between different cloud vendors (Prasad et al. 2010). In simple terms, different types of cloud technologies must have the ability to function and interact together. These technologies including hypervisors, VM technologies, storage and management interfaces. Currently, each cloud vendor has its own way on how cloud clients/applications/users interact with cloud, and it causes interoperability issues when collaborating different cloud technologies. Many of these challenges are confronted with technological tools, which are employed for implementation of services whereas other challenges are related to technical aspects and may be related to management decisions (Rashidi et al. 2013). As an example, when an application owner needs to migrate a Webserver from a VMWare based private cloud to a Xen based Amazon EC2 datacenter, it has to customise some of its configuration (OS, management tools, virtualisation format, VM configuration, storage system, and networking environment) to be able to be compatible with the target environment.

Heterogeneous APIs (Application Programming Interfaces) of different cloud vendors are creating interoperability issues in cloud. Unfortunately, most of APIs across multiple public and private cloud IaaS are incompatible. They tend to have proprietary APIs which are not designed for cross-cloud interoperability. Despite plenty of discussions, cloud still has few

standards and cloud developers like to follow their own ideas on how a cloud and the applications within it should be accessed, operated and managed.

2.5.4 Vendor Lock-In

Companies are reluctant to be tied down a single cloud provider. Vendor lock-in is a problem in cloud computing, it's a situation where cloud users are dependent on a single cloud provider technology and cannot move easily to another provider without substantial cost, legal constraints or technical incompatibilities. The two main reasons are the lack of world-wide adopted cloud standards or interfaces to support portability and interoperability among cloud vendors.

2.6 The Future of Cloud Computing

In the initial years of cloud computing, researchers were curious about the definition and development of cloud, and investigated how they can apply the new cloud computing solution to current issues. Since 2000, researchers have examined cloud computing delivery models and investigated the benefits and risks of cloud computing. Most researchers agreed that cloud computing creates a significant benefit to enterprises in terms of cost reduction. The next generation of cloud computing will focus on abstracting virtual infrastructure and the operational process that go along with the managing that infrastructure .

Interoperability Multi-Clouds

Interoperability will be the core factor which can make cloud computing a hit in the market. Cloud providers are working on interpretability of IaaS and PaaS clouds, then an organisation will be able to use multiple public clouds from different vendors and avoid vendor locking.

Geo- Location Assurance

Future SLA may restrict the Cloud Service Provider (CSP) to keep data in a particular geographic region at the granularity level of city, state, time zone, or a political boundary (Zafar, 2017). CSP will be legally bounded to disclose geolocation of data in the SLA as a requirement. It needs a legal mechanism to regulate dishonesty or violation of SLA by CSP, and also cloud users must have a mechanism to detect these types of violations.

Google NEXT Project

According to an article which was published by Mike Kavis (2016), Google held a NEXT conference in San Francisco, 2015 to show how serious the company is about cloud computing to win enterprise customers. Google is planning to invest a huge amount of money and resources on building of world-class data centres that are equipped with high security, high performance computing and scalabilities. Google is planning to introduce new tools that will be used in a cloud computing environment. The big data is a hot topic these days, and now Google wants shift the conversation from big data analytics to Deep Learning. Deep learning is based on machine learning and google believes machine learning is the next layer of programming. Its goal is to make the process of data ingestion, storage, and training machine models as simple as calling an API (Kavis, 2016). Google also introduced speech, sound, and image recognition services, and the Google Cloud Vision API can analyse an image and categorize its content into thousands of categories.

Cloud Migration Services

With the maturation of cloud, cloud migration applications will be popular and a hot deal. The competition between the cloud players will be on the basis of features and pricing. Companies will prefer CSPs that offer new features, affordability, and more widespread functionalities. The positive part of this being that a company can migrate its data to another provider easily, migration from private to public or vice versa. It will increase the demand for organised migration services with a lift-and-push approach.

Cloud Applications Boom

In the coming two years, developers will recognise that their struggle to meet the speed and functionality of in-house infrastructure is over. Having access to already built public PaaS and IaaS, developers will realise that their major task to deliver applications rather than spending time to improve infrastructure and other functionalities. IT organisations will adopt the fastest infrastructure available and focus on delivering a new class of applications.

2.7 Chapter Summary

This chapter presented an analysis of cloud computing, which has come a long way over the last several years. Cloud computing adopted several existing technologies such as, Distributed Computing, Grid Computing, Utility Computing, and Autonomic Computing to develop the cloud computing concept. In the heart of the Cloud Computing infrastructure, a group of reliable services are delivered through powerful data centres that are based on modern virtualisation technologies.

Virtualisation is the foundational element of cloud computing, and it supports simultaneous execution of diverse tasks over a shared hardware platform. Virtualisation is the key enabling technology allowing the creation the creation of an intelligent abstraction layer which hides the complexity of underlying hardware or software. There are different types of virtualisation; Operating System Virtualisation emulates all the components of a guest operating system. Server Virtualisation enables to construct multiple virtual servers (virtual machine) on a physical server. These virtual servers/VMs can be built by dividing physical server in term of hardware and software.

Innovation is increasing at rates never experienced before and will continue to do so. Cloud computing has been matured over the last several years and it offers numerous opportunities for IT companies and other enterprises. Cloud services such as SaaS, IaaS, and PaaS have revolutionized the IT industry. In the traditional model, users had to build the infrastructure or a platform to develop or run their own applications. As an example, a SaaS user does not pay for the software itself, instead, it operates on a rental basis. The user has the authorisation to utilise it for a period of time and pay for the software that they are using, thus avoiding paying large sums of money to obtain the software licence.

Cloud services are mainly categorised on the basis of access, size and proprietorship. There are three main cloud deployment models which are, Public Cloud, Private Cloud and Hybrid Cloud. Public Cloud operates by the service provider, and the infrastructure will be in the premises of the provider. The user doesn’t have any control over the infrastructure or the location. Private Cloud, as the name suggests, it is solely owned by a particular business organisation, enterprise or institution but it can be hosted internally or externally. Hybrid Cloud contains the features from both Public and Private Clouds, and many businesses have already adopted the hybrid solution because it allows them to host their critical business

information on their own infrastructure. It allows their employees to access business critical applications and information, and also offers safety, scalability, and performance.

Even though cloud computing provides many opportunities, there are some challenges to be dealt with if cloud needs to progress to the next level. Some of the main challenges are interoperability, security and data governance.

Cloud computing started as a way to abstract physical infrastructure and data centres. The next generation of cloud will use novel and useful applications such as Google Cloud Vision API, and use complex technologies such as Google Deep Learning which is a machine learning technology. According to an article on Forbes, a long time investor and cloud expert Byron Deeter is predicting that 42 public cloud providers will pass the $ 500 billion mark by 2020 (Konrad, 2015). Deeter also believes that the rise of Amazon Web Services is allowing cloud companies to grow much faster, these predictions provide evidence that the cloud future is bright. The next chapter presents a complete taxonomy of live VM migration in Cloud.

Chapter 3: A Survey of Live VM Migration in Cloud

The objective of this chapter is to give an overview and depth analysis into the live VM migration concept. This chapter starts with an introduction, and then it moves to live VM migration concept where it discusses the live VM migration concept in detail. In that section, it also discusses VM disk state migration, VM memory transferring techniques, and different types of VM migration strategies. Then it moves to performance, cost and metrics where it discusses VM migration cost and different types of performance evaluation metrics. Later part of this chapter, it discusses properties and challenges of VM migration and VM live migration implementations in commercial cloud platforms, and then the chapter concludes with a summary.

3.1 Introduction

Virtualisation is the key technology behind Cloud Computing, it enables multiple and secure virtual servers to run on a single physical server. Virtualisation technology was implemented on an IBM mainframe in 1960, it works by adding a special software layer which is called a "Hypervisor" on top of the hardware platform. Virtual machines run on top of the hypervisor which provisions hardware resources to the VMs.

A virtual machine concept has been in the IT community since the 1960s where system engineers and researchers at MIT (Massachusetts Institute of Technology) recognized the need for virtual machines. A group of researchers at MIT started working on a project call "Compatible Time Sharing System" (CTSS) and the papers discussed the concept were began to publish in 1959 (Varian, 1997). The purpose of the project was to develop a Time Sharing System (TSS) to allow project teams to use part of the mainframe computers. CTSS ran on MIT's Project MAC (the project on mathematics and computation) but the system did not become popular (Varian, 1997).

Of IBM's (International Business Machines) CP-67 was the first commercial main frame to support virtualisation. The operating system that ran on the CP -67 was CP/CMS, CP stands for Control Program and CMS stands for Control Monitor System. The CTSS system on a

CP-67 main frame allowed each user to have their own complete operating system which effectively gave each user their own computer resulting in a much simpler operation.

While earlier CP/CMS software was made available to customers, according to the Melinda Varian, IBM and MIT had collaborated to develop the IBM OS /360 model. The IBM announced its official first VM, the release 1 of VM/370 was shipped to customers in 1972 and the most important new function in VM/370 release 1 was the ability to run VM under VM (Varian, 1997). The VM/370 was a VMM that ran on the System/370 Extended Architecture (370-XA), which provided specific CPU instructions designed to maximise the performance of running virtual machines (Rose,2004) .

According to Ameen, the researcher Goldberg proposes the “Hardware Virtualizer”, in which VM would communicate directly with the hardware. Later Intel and AMD have revealed specifications for Pasifica and Vanderpool chip technologies that equipped with special virtualisation features (Ameen, 2013). Since the released of first VM in 1972, VM concept has come a long way, different types of virtualisation has been thoroughly analysed and explored in the Chapter 2.

Server virtualisation is a technique that changes server model architecture and establishes a new layer called hypervisor between the physical layer and operating systems. This layer includes several blocks of activities which are controlled by a management system (Ahmadi et al, 2010). Typically in an enterprise environment, it deploys hundreds of servers in a datacentre, server virtualisation attempts to increase resource utilization by partitioning physicals servers into several multiple virtual servers, each running its own operating system and applications. The server virtualisation architecture is shown in the figure 3.1.

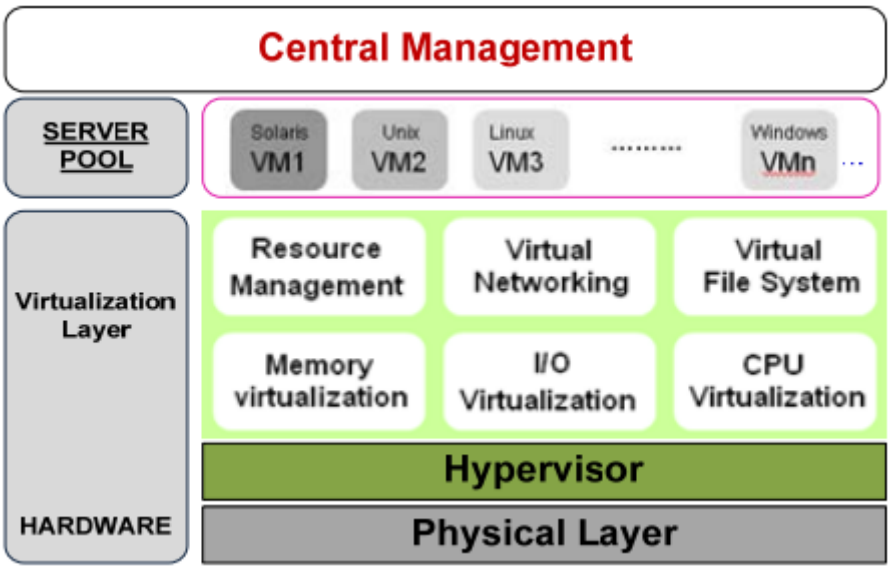


Figure 3.1: Server Virtualisation Architecture (Ahmadi et al.2010)

VM can be defined as "A system which is a hardware-software duplicate of the real machine, in which a non-trivial subset of the virtual machine's instructions executes directly on the host machine (Ameen, 2013).

The capability of moving a VM across different hosts without disrupting the VM's applications and its users provides a significant benefit for administrators of cloud data centres. Live VM migration is an extremely powerful tool in many key scenarios:

1. Load balancing
2. Online maintenance and proactive fault tolerance
3. Power management

In such situations, the combination of virtualisation and VM live migration significantly improves manageability of data centres and clusters.

In each solution, there are three main states that should be transferred: VM's physical memory, the network connections plus the virtual device state, and the SCSI storage. The most intractable issue is migrating the physical memory of the VM, because this is the main factor that affects the migration downtime. In this research, only the VM memory transferring is considered.

There are two main memory transferring techniques, which are pre-copy and post-copy memory transferring techniques. According to Liu et al, pre-copy memory transferring technique decreases the migration downtime by a magnitude of milliseconds (Liu, 2009).

3. 2 Live Virtual Machine Migration Concept

Live VM migration is a technique that migrates the entire system of a VM, including OS, memory, storage, process and network resources and also its associated applications from one physical machine to another without disrupting the client or running applications (Leelipushpam et al. 2013). There are three main physical resources should be considered when performing live VM migration, these are memory, disk and network resources. There are two main methods of migrating a VM, offline migration (cold VM migration) and online/live migration (hot VM migration). In offline VM migration, the services running on VMs are completely stopped during the migration process while the live migration method keeps all services running on VMs.

According to Mahdi et al (Mahdi et al. 2014) there are several steps involved in the live VM migration processes, which are summarised below.

1. **The Setup Stage** – In this stage, it starts the VM migration process starts by selecting the VMs that need to be migrated along with the destination physical machine. It sets up a TCP connection between the source and the destination host in order to transfer VM memory and configuration data. On the destination physical host, the skeleton of the VM is set up and the required memory portion is allocated to the VM.

2. **Memory Transfer Stage** - This stage involves transferring host VM's memory to the set up VM on the destination physical host. There are several VM memory transferring methods, but it is essential to select a method that offers short VM downtime.

3. **VM Storage Transfer Stage -** This stage involves transferring the virtual storage of the source VM to the destination VM.

4. **The Network Clean Up Stage** – It is important to keep open all network connections that were open before the VM migration in order to perform smooth VM migration. Each VM has a Virtual Network Interface Card (VNIC), which is identified by a

unique MAC (Media Access Control) address. The VM needs to update its new VNIC to the switches in the network in order to forward the traffic to the new VM.

Live migration brings many benefits such as energy saving, load balancing, and online maintenance. Using live VM migration, the VM consolidation can be archived in a cloud based data center by moving VMs from under utilised servers to other servers and shutting down unused servers. Live migration also helps in implementing fault tolerance by migrating the VM from failing physical machines.

3.2.1 VM Disk State Migration

Disk state migration of VM is another important factor of VM live migration; it involves transferring the VM's virtual hard disk from source to the destination host. The virtual machine consists of one or several virtual hard disks; the VM stores its operating system, programs and other data files on its virtual hard disks, these virtual hard disks are also needed when live VM migration occurs. Early live VM migration solutions did not migrate the virtual disks of the source VM but only the state of the CPU and its main memory. Instead, it is required that virtual disks of VM reside on the same shared volume accessible by both source and destination host (Mashtizadeh et al.2013). Live storage migration overcomes this limitation by enabling the movement of virtual disks across storage elements. This means, VM disks can be migrated from one storage domain to another while the VM to which they are attached is running. Multiple approaches to live storage migration are possible and each offers different trade-offs by way of functionality, implementation and performance (Mashtizadeh et al.2013).

According to Mashtizadeh et al (2013), there are three main live storage migration techniques in VMware ESxi that migrate VM disk images without service interruption. The techniques are, Dirty Block Tracking (DBT), IO Monitoring and Snapshotting. DBT is a widely adopted technique by many VM vendors (e.g. Xen and VMware ESX), it uses bitmap to track a write request while the VM image is being copied. Once the entire image has been copied to the destination, a merge process is initiated to patch all the dirty blocks from the original image to the new, the disadvantage of this technique is if a number of dirty blocks are not converged due to a heavy write request, the migration of VM would be significantly long (Zhou et al.2013).

Disk state migration strategies may vary depending on their migration environment (LAN or WAN) and storage methods. Most modern datacentres are equipped with Network Attached Storage (NAS) or Storage Area Network (SAN) storage methods. In a virtualised environment, NAS and SAN storage can serve as a swap space to move VMs between servers. And also they can act as a backup medium or play the role of central repository for all virtual disk images (Network Evolution.2011).

In NAS, the storage can be accessed uniformly from all machines in the cluster and this advantage avoids the need to migrate disk storage (Lie et al, 2011). In a WAN environment, it is hard to transfer the disk state due to the network connectivity holding and bulk storage replication, storage migration over WAN faces significant performance challenges. However, VM disk state migration is beyond the scope of this research.

3.2.2 Virtual Machine Migration Techniques

There are three different categories of migration techniques as follows, (Leelipushpam et al. 2013)

- Load Balancing Migration
- Energy Efficient and VM Consolidation Migration
- Fault Tolerance Migration

3.2.2.1 Load Balancing Migration

Cloud data centers are highly dynamic and unpredictable due to irregular resource usage patterns of cloud users. Load balancing is one of the main challenges in a datacenter that equipped with hundreds or thousands of VMs in order to ensure equal utilisation of all available resources while avoiding a subset of machines.

In Load Balancing the VM technique, the aim is to distribute load across all physical servers in order to avoid bottlenecks, improve availability and over provisioning of resources. The below figure 3.2 shows that the virtual machine 3 of physical server 2 is migrating to physical server 3 to balance the load on the physical server 2.

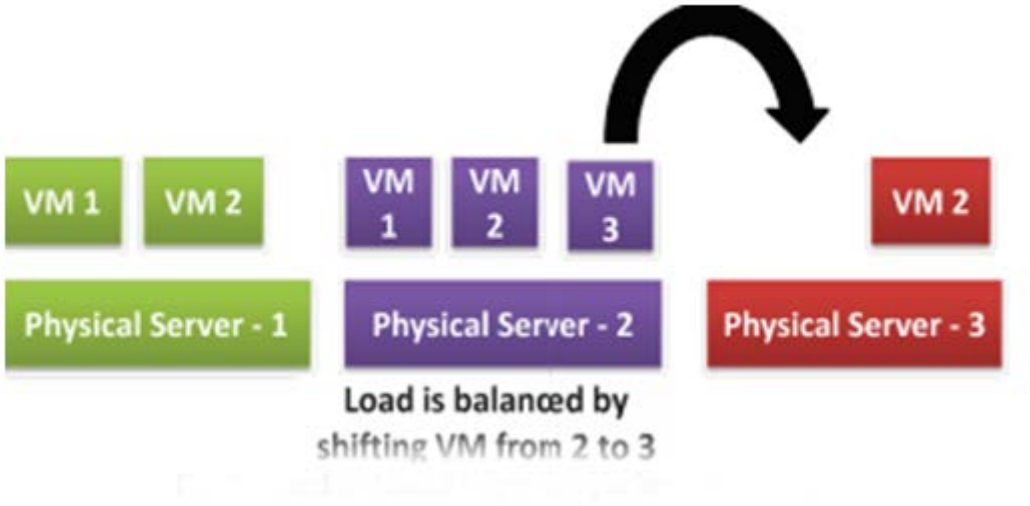


Figure 3.2: Load Balancing Migration (Leepushpam et al. 2013)

Traditionally, there are several ways to achieve load balancing in a networked system, one simple technique is static load balancing, but efficiency of this method depends on the accuracy of the prior predication (Barbalace et al.2014). Dynamic load balancing strategy can be archived by executing VM migration process in a dynamic manner by mapping of jobs to physical resources. Deciding underloaded and overloaded machines and VM migration will be triggered by looking at usage of resources by physical machines in real time.

3.2.2.2 Energy Efficient and VM Consolidation Migration

The resources of virtual machines are limited to its underlying resources of the hosting physical machine. VM migration allows several VMs to move into a single physical server using a technique called VM consolidation. The main idea behind the VM consolidation is executing VMs on as few physical machines as possible to concentrate the workload and efficiently limit the number of physical servers with power on (Corradi et al.2014).

The main idea behind this technique is to conserve energy by migrating VMs from underutilised servers to servers that have enough capacity to host them, and then underutilised servers can be shut down to save energy in the data centre. In the figure 3.3 shows that physical server 2 is underutilised and VM1 on physical server 2 has migrated to the physical server 1, then physical server 2 can be shut down to conserve energy. Figure 3.3 shows the energy efficient migration technique.

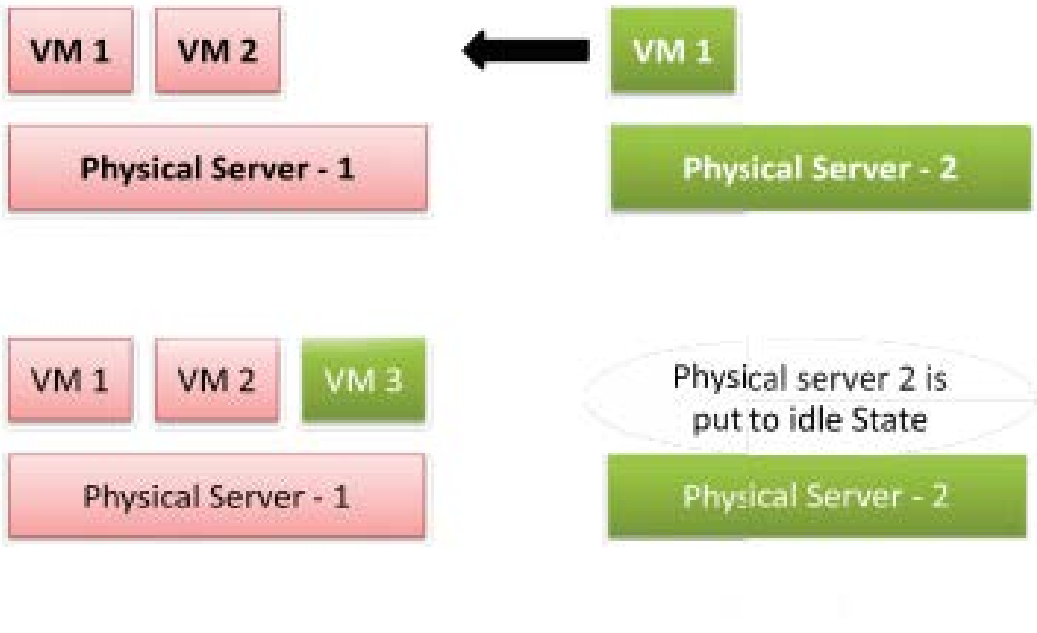


Figure 3.3 – Energy Efficient VM Migration Technique (Leepushpam et al.2013)

3.2.2.3 Online Maintenance and Proactive Fault Tolerance Migration Technique

This technique aims to predict a physical machine failure beforehand and migrate all VMs from the failing physical server to another physical server, this technique improves availability in cloud based data centres. Figure 3.3 shows that the physical server 2 is about to go down, then the VM1 and VM 2 have migrated to other physical servers in order to maintain availability. Figure 3.4 shows VM1 and VM2 migrating from the failing server 2 in order to provide continuous service for the users on VM1 and VM2.

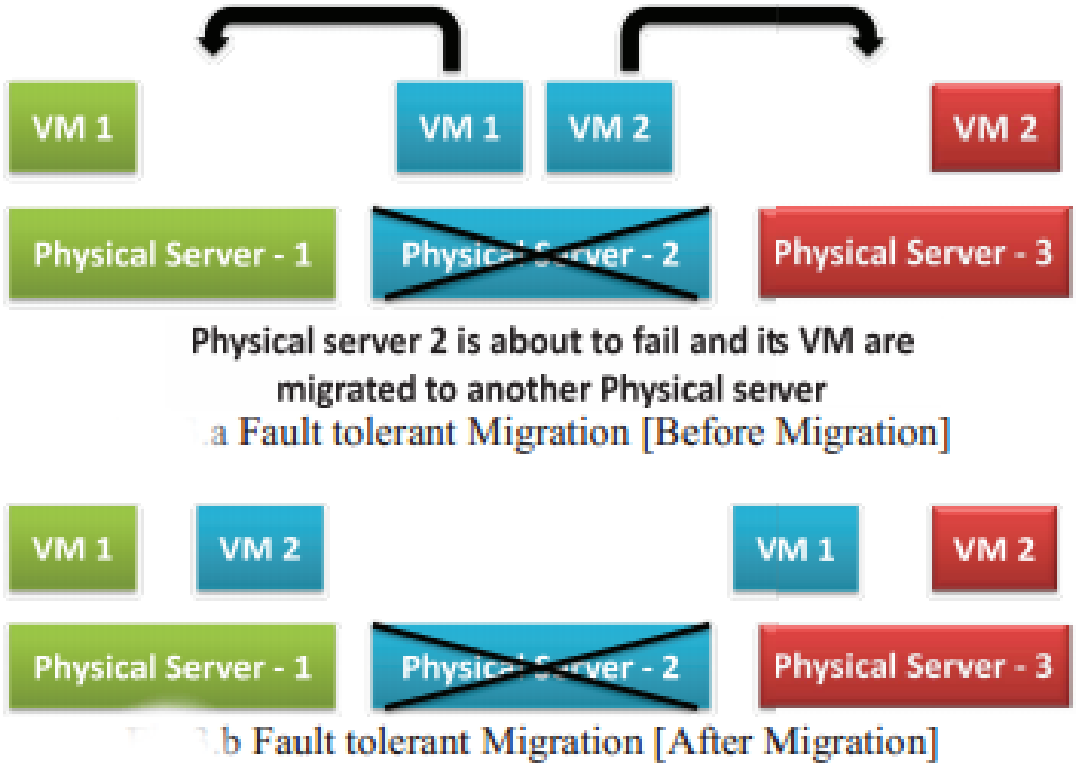


Figure 3.4 – Fault Tolerance Migration (Leepushpam et al.2013)

3.2.2 Virtual Machine Memory Migrations

In live VM migration, the most important phase is transferring the source VM memory to the destination VM. Memory migration can be divided into three phases, push phase, stop-and-copy phase and pull phase. According to Botero (Botero,2012) in push phase, the source continues running while certain pages are pushed across the network to the new destination. In the Stop-and-copy phase, the source VM stops, pages are copied across to the destination, and then the new VM starts. In the pull phase, the new VM starts its execution and if it accesses a page that has not yet been copied, then the page is copied across to the destination. Most migration strategies select either one or two of the above phases, the pre copy approach combines push with the stop-and-copy phase while the post-copy approach combines pull copy with stop-and-copy. (Botero, 2012).

3.2.2.1 Two Main Memory Migration Methods

There are two major approaches in VM memory migration, pre-copy and post-copy VM migration.

Pre-copy VM Migration

The Pre-Copy method is used in many commercial hypervisors such as KVM, and ZEN, there are two phases in the pre-copy method, the warm-up phase and stop-and-copy phase.

The process behind the pre-copy VM migration is to transfer the virtual machine's memory from the host to the destination over a series of iterations. This technique iteratively copies the memory pages from the source machine to the destination host without stopping the VM. This method needs irritation of memory transfer due to the dirty pages. The dirty pages are the pages that have been modified since the last page transfer, and they need to be sent to the destination host until the rate of recopy pages are not less than dirty page rate. The process continues until the writable work set (WWS) becomes small. When the WWS rate becomes small, it performs stop and copy then transfer of the dirty pages to the CPU (Agrawal, 2013).

According to Patel et al. (2014) the steps of Pre-copy VM migration process are shown below.

1. Warm-up phase – Copies all the memory pages from the source VM to the destination host while the VM is still running. If some memory pages change during the memory copying process (dirty pages) they need to be copied until the rate of recopy pages is not less than dirty page rate.

2. Stop and Copy Phase – In this stage, the VM on source destination will be stopped and remaining dirty pages will be copied to the destination and the VM will be started in the destination host.

Post-Copy VM Migration

In post-copy VM migration, the VM is immediately suspended upon starting of the VM migration. Then it copies the minimum execution state to the destination, and the VM is resumed at the destination host. The VM's memory is both actively pushed from the source and fetch paged from the destination. Post-Copy method has low network overhead as it transfers dirt pages only once, but the VM may experience huge performance degradation till all of its working set pages and demand pages pushed from the source.

The Steps of Post-copy VM Migration

In this approach, post-copy suspends the migrating VM at the source machine, copies the minimal CPU state to the host, then resumes the virtual machine at the target node. After that it begins fetching memory pages over the network from the source.

According to Hines et al, the steps of post-copy VM migration (Hines et al.2009) are shown below.

1. Preparation Time – This is the stage between initiating VM migration and transferring the VM's CPU state to the target host. In this stage the VM continues to execute and dirty its memory.
2. Downtime – In this stage, migration VM execution is stopped.
3. Resume Time – This is the stage between resuming the VM's execution at the target host and the end of migration altogether.
4. Pages Transferred – This is the total amount of VM memory pages transferred including the duplicates.
5. Total Migration Time - This is the sum of above all stages, the total time is important as it affects the release of resources on both source and target hosts.
6. Application Degradation – At this stage the applications that are running on the VM might be slowed down

A simplistic model for VM memory transfer has been implemented by Moghaddam (Moghaddam et al. 2011), this model illustrates the parameters that participate in a typical VM live migration. According to the model we assumed that r is VM RAM, b is the link bandwidth, d is delay and error rate is err and the memory transfer time is t_d .

$$t_d = \frac{r}{b}(1 + err) + d$$

As an example, if we assume RAM (r) = 8 GB, b = 1GB/Sec, err = 0.1 %, d = 0.5 sec, and the memory transfer time will be 64.564 Sec.

3.2.3 VM Memory Migration Categories

To help understand the VM memory migration, VM memory is categorised into five major categories (Hu et al. 2014).

1. VM Configured Memory
2. Hypervisor Allocated Memory

3. VM Used Memory
4. Application Requested Memory
5. Application Actively Dirtied Memory

VM configured memory is the amount of memory that has been allocated to the virtual machine by the hypervisor. The virtual machine's guests are seeing this memory as their "physical memory" that is available for use. Allocated memory is the amount of physical memory on the underlying hardware that the hypervisor has actually allocated to guest VM. VM used memory is the amount of memory which has been currently and actively used by VM's operating system. Application requested memory is the amount of memory requested by the applications that are running on the virtual machine.

Dirtied memory is the memory that VM's application are actively modifying via writing to in-memory pages. Application dirtied memory is also a part of the Requested Memory of an application.

3.3 Live VM Migration Performance, Cost and Metrics

Live VM migration is a collective of several iterations, transfer of VM memory, storage, and network resources are part of this process. The most important aspect of live VM is transferring VM memory, and multiple iterations are required to transfer the VM memory, it can be two types, a non-adaptive and adaptive. In the non-adaptive process, VM memory pages are transferred at the maximum possible rate, in the adaptive process, the rate of the memory page transfer is tied to the dirty page rate (Nethan et al. 2013).

The performance of any live VM migration strategy could be gauged by the following metrics. (Patel et al.2014).

1. Preparation Time: Time between initiating the migration process and transferring the VM CPU state to the destination host.
2. Down Time: VM execution is stopped completely.
3. Page Transferred: Total amount of memory pages transferred including duplicates
4. Resume Time: This is the time between resuming the VM at the host destination and the end of the VM migration.
5. Total Migration Time: The total time taken to complete the whole migration process.

6. Application Degradation: When a VM migrates from source to destination, it degrades the performance of running applications on VMs.

The total migration time and the downtime are the two major metrics that are used to evaluate the performance of live VM migration. The total migration time is the amount of time needed to complete the VM migration process, and the downtime is the interruption period of the services running on the migrating VMs (Zhang et al. 2014).

The main difference between two methods is the pre-copy method, which transfers VM memory pages without suspending the source VM, but in the post-copy method it first suspends the source VM to transfer the minimum amount of VM memory needed to start the VM in the destination host.

The cost of the live VM migration depends on all the below mentioned parameters, each parameter can impact the performance of applications that are running on VMs. The cost of live VM migration process can be evaluated in terms of the following factors (Waltenegus, 2014)

- The memory content and the memory update rate of each virtual machine.
- The total number of virtual machines to be migrated.
- The available network bandwidth for migration.
- The workload of the source and the destination servers at the time of migration.

According to the Waltenegus experiment result, it was revealed that power consumption of both the source and the destination servers proportionally increased as the available network bandwidth decreased. The power consumption of the server however was not affected by the RAM size of migrating VMs. The total VM migration time decreased as the network bandwidth increased. The VM migration time increased linearly as the RAM size of the virtual machine increased (Waltenegus, 2014).

3.3.1 Properties and Challenges of Live Migration

All live migration approaches aim to full fill the fundamental criteria of the live VM migration concept. According to Peter Svard et al, the fundamental aspects of live migration are summarised in a number of properties as presented below (Svard et al.2014).

1. Continuous Service: The live VM migration process should not disrupt users or experience performance degradation.
2. Low resource usage: The live VM migration process consumes resources on source and at the destination host, mainly memory, network bandwidth and CPU resources. The resources consumption must be kept at a minimum during the live VM migration in order to prevent SLA (Service Level Agreement) violations.
3. Robustness: The hypervisor, the VM, and hosted application should not risk crashing or freezing due to the live VM migration process.
4. Predictability: It should be possible to predict the duration of the migration process and resources consumption.
5. Transparency: The migration process should be transparent to both VM and its users.

Any live VM migration algorithm must fulfil the above five properties to be an ideal algorithm, but this is not possible in practice.

Live Migration Challenges

There are parameters that hamper live migration, which are the transfer rate problem, the page re-send problem, and the missing-page problem (Svard et al.2014).

The transfer rate problem, during the iteration process of pre-copy live VM migration, the VM continues to run and its memory content is constantly updated. This problem occurs, when memory pages are being dirtied at a faster rate than the page transfer rate over the network. This means the VM memory transferring process gets stuck and as a result the migration process may be forced to proceed the stop-and-copy phase. This leads to extended migration downtime and a prolonged total migration.

The page-resend problem, live migration consumes a significant amount of CPU, memory and network bandwidth as it transfers several gigabytes of VM memory during the live VM process. This problem is simplified in pre-copy migration as the source VM is running during the iterative phase and pages that have already been transferred are often being dirtied again (Svard et al.2014). In order to re-start the VM at the destination host, an exact copy of the source VM state needs to be copied to the host server, so the dirtied pages need to be re-sent to the host.

The missing pages problem, in post-copy live migration, the live VM migration process resumes the VM at the destination host before it is complete and the memory contents have been transferred to the destination host. After the execution is switched to the destination

side, the missing pages are pulling over the network from the source. However this process adds extra strain on the network as it consumes a significant amount of bandwidth. This problem is referred to as the missing pages problem, and it imposes a high risk of performance degradation for physical machines, VMs and VM applications.

Security- Live VM migration is a novel idea and the security aspect of it is not fully discovered. Anala et al demonstrated security threats for live VM migration. According to them, live VM migration can be attacked in one of the three following classes (Anala et al.2013).

1. Control Plane
2. Data Plane
3. Migration Module

A lack of security in the control plane may allow an attacker to exploit live migration operation in different ways. As an example, an attacker can create many VMs on the host Operating System (OS) to make the OS overloaded. In the data plane class, VM memory transferring is prone to a man-in-the-middle attack because when VM memory is moving from one host to another, and the data are not encrypted and an attacker may place himself to perform a man-in-the middle attack.

3.4 Live VM Migration Implementations in Commercial Cloud Platforms

Live VM migration offers many benefits for cloud users and platforms with live VM migration offering redundancy, therefore having the live VM migration as a feature in a cloud platform is a plus point. In this section, it will discuss some of the popular cloud platforms that offer live VM migration as feature.

3.4.1 Google Compute Engine (GCE)

Google announced Google Compute Engine (GCE) on 28th of June, 2012 in a limited preview mode. The live migration process in GCE platform transfers a running VM from one host machine to another host within the same zone. When live VM migration occurs, all VM properties and attributes remain unchanged including internal and external IP addresses,

instance metadata, block storage data and volumes , network connections and settings and so on (Cloud Google, 2017) .

In GCE, the live VM process begins with a notification sent to the VMs that needs to be migrated from their current host machine. The notification might be sent due to one of the following reasons,

- Regular maintenance and upgrades.
- Network and power grid maintenance in the data centre.
- Failed hardware.
- Host OS and BIOS upgrades.
- Security upgrades
- System configuration changes

Google cluster management software constantly monitors for these events and schedules the VM migration based on policies that control the data centres. When GCE migrates a running VM, it transfers the complete instance state from the source to destination host in a way that is transparent to the guest OS without interrupting any users communicating with it. There are three stages in VM migration in GCE, and it uses a hybrid memory transfer technique (combination of pre-copy and post-copy).

Pre-migration brownout – In this stage VM is still executing on the source, but most of the state is sent from the source to the target. Actually, this is the pre-copy memory transfer phase, but GCE has given it a simple term that users understand.

Blackout Stage – In this stage, the VM is paused for a moment and the processor state is copied to the destination without the remaining dirty memory pages.

Post migration burnout –. In this stage, the post-copy algorithm kicks in and then the VM resumes at the destination immediately. Then it copies the rest of the source memory to the destination host. When the migration completes, the system deletes the source VM.

3.4.2 VMware vSphere Cloud Platform

In 2009, VMware introduced its cloud platform called vSphere, later they released vCloud Air which is the private cloud platform that is built on vSphere. VMware offers Private,

Public and Hybrid cloud solutions, the Hybrid cloud version is called Cross-Cloud architecture.

Live Migration on vSphere

According to VMware, VMware vSphere offers live migration between different hosts (VMware,2017). It allows a user to move an entire running virtual machine from one physical host to another with zero downtime. It transfers VM active memory, and the execution state runs over a high speed network and the entire process takes less than two seconds on a gigabit Ethernet network. The virtual machine retains its network identity and connection after the migration process.

VMware vSphere offers automated VM migration through a component called vMotion, it allows users to schedule VM migration at predefined times without administration intervention. The vMotion allows you to (VMware.com, 2017):

- Migrate multiple virtual machines running on any operating system and across any type of hardware and storage supported by vSphere.
- Identify the optimal placement for a VM in seconds.

3.4.3 OpenStack Cloud

According to Redhat.com, OpenStack is a combination of open source tools (known as projects) that use pooled virtual resources to build and manage private and public clouds (redhat.com, 2017). Six open source tools, compute, networking, storage, identity, and image services are bundled together to create a unique and deployable cloud. However, not all hypervisors support live migration in OpenStack, KVM, QEMU, XenSever/XCP are some of the currently supported hypervisors (Igbe, 2013).

OpenStack Live Migration

There are three live VM migration categories in OpenStack, which are,

- Block live migration without shared storage
- Shared storage based live migration
- Volume backed VM live migration

Block live migration does not require shared storage among nova-compute nodes, and it uses Transmission Control Protocol (TCP) to copy VM disk from source host to destination host. This drawback of this method is it takes a long time to complete the migration and also host performance will be degraded in a network and CPU point of view (Zhang, 2013).

In a Shared Storage Based Live Migration method, VM disk (storage) is stored on a shared storage. Live VM migration is easier since the VM disk is stored on shared storage as it can easily move from one host to another.

In Volume backed VM live migration, instances are backed by volumes rather than a VM disk and shared storage is not needed. VMs are booted from volume and can be easily migrated between hosts (Zhang, 2013).

3.5 Chapter Summary

This chapter started with an introduction beginning with a brief description of history, key technologies behind Cloud Computing, and three main strategies in live VM migration. There are three different types of VM live migration strategies which can be applied to address different issues and purposes in cloud datacentre. The three different categories of migration techniques are: Load Balancing, Energy Efficient Migration, and Fault Tolerance Migration. The ability to move a VM from a physical machine to another host without any downtime takes away the one of the biggest headaches from an IT administrator. IT administrators can perform server maintenance without disrupting the users of VMs. Live VM migration can be used as a Load balancer, it reduces the inequality of resource usage levels across all physical machines in cluster. This prevents some machines from getting overloaded. And also sever consolidation can be achieved by moving VMs and packing as much as possible on a PM, so that the resource usage is improved and then underutilised machines can be shutdown to save energy. It is only load balancing migration technique that has been considered in this research

A comprehensive analysis of the VM live migration concept has been presented in this chapter. There are three key main physical resources must be considered when moving a VM from source to destination, which are VM memory, disk and network resources. There are several steps involved in live VM migration, the setup stage, VM memory transfer stage, VM storage transfer stage and the network clean up stage. Multiple approaches to live storage migration are possible and each offering different trade-offs by way of functionality,

implementation and performance. There are two main live storage migration techniques that migrate VM disk images without service interruption. The techniques are Dirty Block Tracking (DBT) and IO Monitoring. DBT is a widely adopted technique used by many VM vendors (e.g. Xen and VMware ESX).

Most datacentres used NAS (network area storage) or SAN (storage area network) storage techniques and they offer significant advantages. In NAS or SAN storage techniques, there is no need to transfer the disk storage of a VM over a LAN (local area network) , but transferring disk storage of a VM over WAN attracts more attention from researchers (Zhang et al.2017). However, VM disk state migration is beyond the scope of this research.

The most important part of live VM migration is moving the VM memory content from the source to the destination host. Several VM memory migration techniques have been researched in this chapter, mainly pre-copy and post-copy VM memory migration techniques. The main difference between two methods is the pre-copy method, which transfers VM memory pages without suspending the source VM whereas in the post-copy method, it first suspends the source VM to transfer the minimum amount of VM memory that needs to start the VM in the destination host.

Live VM migration performance evaluation has been discussed in this chapter, the total migration time and the downtime are the two major metrics that used to evaluate the performance of live VM migration. The total migration time is the amount of time needed to complete the VM migration process, and the downtime is the interruption period of the services that are running on the migrating VMs. However every VM memory migration method has a downtime and it ranges from a few seconds to minutes, the migration downtime depends on two factors, which are available bandwidth and the VM memory size.

The challenges of implementing load balancing algorithms in a cloud environment have been discussed, and they depend on following factors: network overhead, performance and the point of failure. Live VM migration in the cloud offers many advantages, such as fault tolerance, server consolidation, and energy conservation, therefore having live VM migration as a feature in a cloud platform is a plus point. However, only a few existing cloud platforms offer this feature, some of the commercial cloud platforms which offer live VM migration are discussed and analysed in this chapter. The next chapter presents dynamic load balancing based on live VM migration, analysis of load balancing algorithms and related work.

Chapter 4: Dynamic Load Balancing Based on Live VM Migration

This chapter starts with an in-depth insight into the using of live VM migration as a Loadbalancer in a cloud computing environment. Then it moves to different types of load balancing algorithms where it discusses centralised and distributed load balancing algorithms. Next, it discusses modeling and scheduling process of load balancing algorithms, and then it moves to evaluation metrics where it discusses different types of evaluation metrics. This followed by comprehensive discussion and analysis on existing load balancing implementations. Finally, the chapter concludes with a critical analysis and summary.

4.1 Introduction

The real time challenges of cloud computing are load balancing and the basic reason for this rapid demand is an increase in the number of users and their demand for cloud services. The Scalability is one of the most important features of cloud computing, it is especially important for the network where it's difficult to predict the number of requests that will be issued to a physical server.

Live VM migration is a concept that moves a VM from a source server to a destination host without interrupting its applications and users. Live VM migration can be used as a load balancer to distribute VMs dynamically across all physical nodes to balance workload ensuring no single physical node is overwhelmed by excessive workload. Live VM migration also can be used to archive fault tolerance and conserve energy in data centers.

VM migration scheduling with load balancing is aiming to assign VMs to a suitable host and balance the resource utilisation within all the hosts, which will prevent an overload of the hosts. Load balancing algorithms can be applied at both application level and at the VM level. At the application level, the load balancing algorithm is integrated into an application scheduler, and the load balancing algorithm can be applied to the VM manager (Xu et al. 2017).

Traditionally, there are several ways to achieve load balancing in a cloud environment. Various load balancing algorithms exist to serve different purposes, but the goal of every load

balancing algorithm is to improve performance by balancing the load among different physical servers according to various resources (CPU, memory, disk storage) and also to achieve higher throughput, maximum response time, and avoiding overload (Shah et al.2015). Load balancing algorithms can be divided into two main categories, static and dynamic.

Static load balancing algorithms move VMs among servers without considering the current status of the system, these algorithms require a prior knowledge of system resources utilisation. It does not consider present status of the load and information about the system while distributing the load. These types of algorithms suit environments where there are few load variations.

Dynamic load balancing algorithms constantly check different types of properties of the server such as network bandwidth, CPU utilisation, memory usage, disk storage and other parameters while distributing the load. Dynamic algorithms respond to the actual current system state in making decisions and move VMs from overloaded servers to an underutilised server dynamically in real time.

4.2 Types of Load Balancing Algorithms

Generally, load balancing algorithms in cloud can be implemented in two ways, as a Centralised system or a Distributed system.

Centralised

In a centralised load balancing system, a central component or the scheduler obtains global information (utilisation, load, connections and other information) from hosts in the cluster. In each execution process of the centralised algorithms, the states of all hosts are collected, analysed and reordered to provide information for VM allocation (Xu et al. 2010). The benefits of a central management load balancing system are that it’s easier to implement, manage and quicker to repair in case of a failure. Centralised load balancing architecture is shown in the figure 4.1.

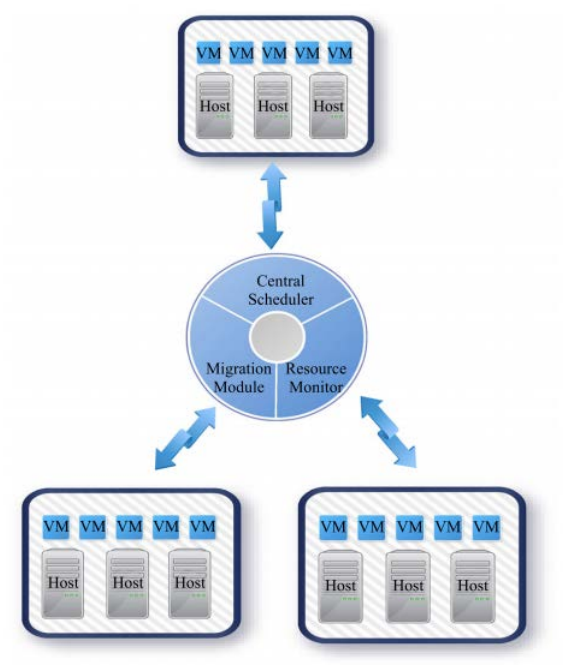


Figure 4.1: Centralised Load balancing architecture (Xu et al. 2010)

Example of a commercial centralised load balancing system is a Red Hat Enterprise Virtualisation Suite (Hildred et al. 2011). The disadvantages of a centralised scheduler are a single point of failure and system bottlenecks due to network overhead.

Distributed Load Balancing Scheduler

In a distributed load balancing scheduler, the algorithms enable the scheduling decisions to be made by the local scheduler on each node. Distributed schedulers are located on each node rather than on a central node, due to that reason the associated network overhead is distributed. The distributed algorithms eliminate bottlenecks and improve reliability and scalability, because they avoid a single point of failure. The distributed Load balancing Scheduler is shown in the figure 4.2.

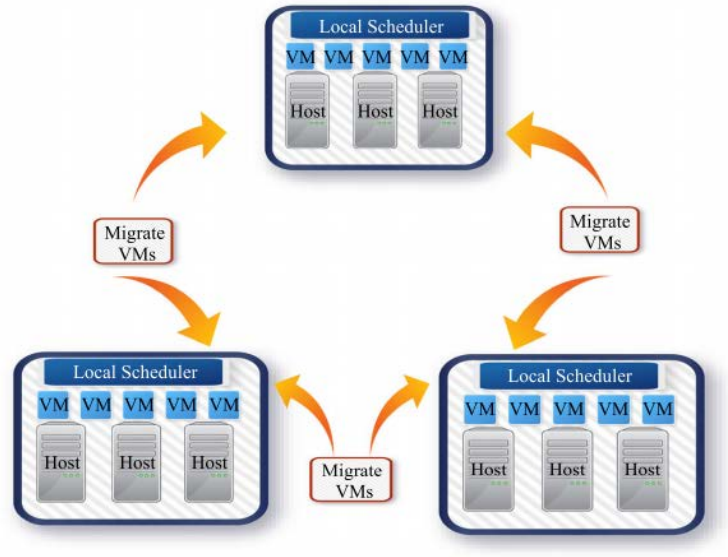


Figure 4.2: Distributed Load balancing Scheduler (Xu et al. 2010)

4.3 Challenges of Implementing Load Balancing Algorithms

The challenges of implementing load balancing algorithms in a cloud environment depend on following factors, network overhead, performance, and reliability.

Overhead

Network overhead determines the amount of overhead involved while implementing load balancing, overhead can be determined by VM migration cost, and communication cost of load balancing algorithms.

Transferring the memory of the VM is one of the most important processes in live migration.VM memory should be transferred to the destination VM before resuming the new VM at the destination host. If the migration is seamless, then users won't face any disruption to their services during or after the VM migration. But unfortunately every VM memory migration method has a downtime ranging from a few seconds to minutes, this migration downtime depends on two factors, being available bandwidth and the VM memory size.

In data centres, VMs are exposed to a huge amount of user traffic and if VM migration occurs on a network link that is overwhelmed with user traffic, both the user traffic and VM migration traffic would compete for the available bandwidth and then create a bottleneck if there is insufficient bandwidth on the link to support the VM migration.

Performance

Performance determines the efficiency of the load balancing system, and it includes following factors, resource utilisation, scalability, and response time. Collecting resources utilisation from hosts and applying load balancing algorithms, it can detect overloaded and underloaded hosts in the cluster, hosts with higher resource utilisation should be offloaded. Load balancing algorithms should achieve scalability and must provide smooth operation of services even if the numbers of users increase, it represents the quality of service. Response time is the amount of time taken to react by load balancing algorithms. Lower the response time, better performance results.

Reliability

Reliability is an important factor when implementing a load balancing system, it should be designed to prevent single point of failure. Like in a centralised system; if the central node fails, then the whole system would fail, so load balancing algorithms should be designed to overcome this issue.

4.4 Modelling of VM Load Balancing Algorithms in Cloud

When designing VM load balancing algorithms in cloud, some basic principles should be considered, which are VM resource type, VM type uniformity, allocation dynamicity, optimisation strategy, and scheduling process (Xu et al. 2010).

VM Resource Types

When designing load balancing algorithms for VM scheduling, single resource type or multiple resource type must be taken into consideration. In a single resource type, it takes only one resource type into consideration for scheduling VMs. Generally, it’s limited to a single resource type, mainly CPU utilisation, this assumption is made to simplify the load balancing process.

Multiple resource type is considered in some load balancing algorithms, which monitors several VM and Server resource types such as CPU, memory, I/O load. These load balancing algorithms take decisions by monitoring combination of several resources utilisation patterns, and configuring different resources with weights or identify different resources with priorities.

VM Machine Uniformity

In VM load balancing algorithms, the VM scheduling is modelled as homogeneous or heterogeneous. In the homogeneous model, load balancing algorithms are designed as all VMs' tasks have the same characteristics; this assumption is made to simplify the scheduling process but it ignores the diverse characteristics of tasks.

In the heterogeneous model, VM load balancing algorithms for VM scheduling is modelled by taking various task characteristics and scheduling objectives into consideration. The reason being cloud providers provide different types of VMs to support various task characteristics.

Virtual Machine Allocation

Load balancing algorithms for VM allocation can be classified as static or dynamic. The static algorithms are also called offline algorithms, in which the VMs information are required to be known in advance (Xu et al. 2010). Dynamic VM allocation algorithms are referred to as online algorithms, in which VMs are dynamically allocated according to the loads at each time interval. These algorithms can dynamically configure the VM placement combining with VM migration techniques.

4.5 Load Balancing Scheduling Process

Scheduling VM migration divides into two main categories, the VM initial placement stage and VM live migration stage. In the VM initial placement stage, the live migration is not considered, instead it focuses on selecting a suitable host for migrating VMs. Generally, host selection algorithms take the host's available resources into consideration.

In VM live migration stage, it mainly considers following aspects.

1. VM migration policies enable cloud providers to establish presences when VMs migrating from one host to another. VM migration algorithms consist with a migration threshold, and it triggers VM migration. VM migration threshold is based on computing capabilities of each host. As an example, CPU intensive host may be configured with a relatively high threshold on CPU utilisation.

2. VM selection policies enable VM selection algorithms to select the best VMs to migrate from overloaded hosts. VM selection policies also determine overloaded and underloaded hosts.

4.6 Evaluation Metrics of VM Load Balancing Algorithms

There are different types of load balancing algorithm evaluation metrics, these metrics are optimised on the basis of different behaviours, like obtaining maximal or minimal values. Following are some of prominent adopted metrics in VM load balancing algorithms.

Load variance and standard deviation of utilisation

Both two metrics specify the deviation from the mean utilisation (Xu et al. 2010), and these are easy to measure.

Makespan

Makespan is the longest processing time on all hosts, and it's one of the most common evaluations metric for VM scheduling algorithms. The primary purpose of a scheduling algorithm is shortening the makespan, and it pays more attention to time constraint. Makespan is favourable in evaluating real time scheduling load balancing algorithms.

Associated Overhead

The amount of overhead that is produced by the execution of the load balancing can be measured (Mishra et al. 2015). Minimum overhead is expected for successful implementation of the algorithm.

Throughtput

It measures the total number of tasks, whose execution has been completed successfully. Therefore, higher throughput comes along better system load balancing situation.

Number of Overloaded Hosts

The metric measures how many hosts in the cluster are overloaded, and which gives an overview of the system status. The value is depending on the reconfiguration of overloaded threshold. Load balancing algorithm aims to reduce the number of overloaded hosts as much as possible.

Average Imbalance Level

This metric considers multiple types of resources together, such as CPU, memory, then it measures the deviation of these resources on all hosts, and then combines them together with the weights to denote the load balance effects (Tian et al. 2015).

Imbalance Score

It represents the degree of overloaded a host based on exponential weighting function, which aims to overcome the limitation of linear scoring (Singh et al.2008). This metric provides reference about how high the host utilisation is above the predefined threshold and also it considers the multiple resources.

Number of Migrations

If load balancing algorithms cause unnecessary migration, it may degrade performance. This is an auxiliary metric that represents the performance and it is measured with other metrics together (Xu et al. 2010).

4.7 VM Load Balancing Implementations

This section will discuss a few existing VM load balancing implementations which are analysed below.

4.7.1 Migration Management Agent

Song et al proposed a VM migration management agent or algorithm to balance the load dynamically in high-level applications (HLA) federations (Song et al.2015). HLA systems, such as large scale military systems, its computation and communication loads may vary and dynamic during their execution time. This algorithm allows VMs to migrate between different federations in order to balance the load. The objectives of this approach are reducing the load of overloaded hosts and decreasing the communication cost among different federations (Song et al. 2015). Figure 4.3 shows the communication architecture of VM/Federate below.

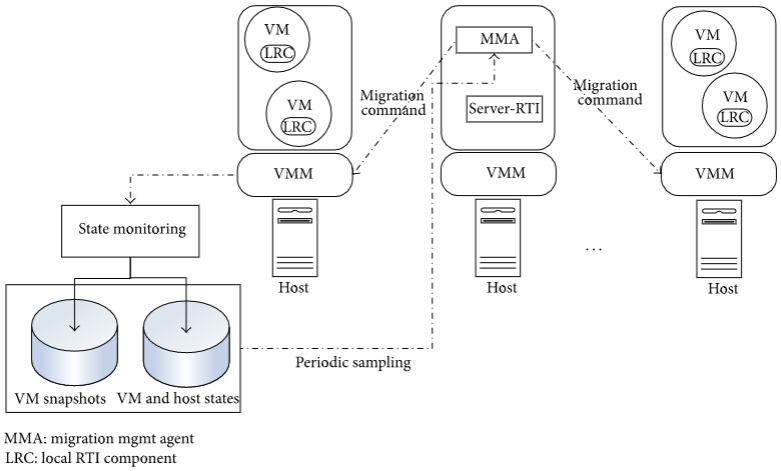


Figure 4.3: VM/Federate Communication Architecture (Song et al. 2015).

The authors predefined host CPU utilisation threshold for detecting overloaded and underloaded physical machines, and VMs. They also modelled communication cost for VMs within the same host and different hosts.

Host Utilization Threshold

U_{pj} *(t)* is the utilisation of host P_j at time *t* :

U_{pj} *(t)* = $\alpha \times U_{cpu}$ *(t)* + *(1* − α*)* U_{mem} *(t),*

Th_p = *k, 0 < k ≤ 1,*

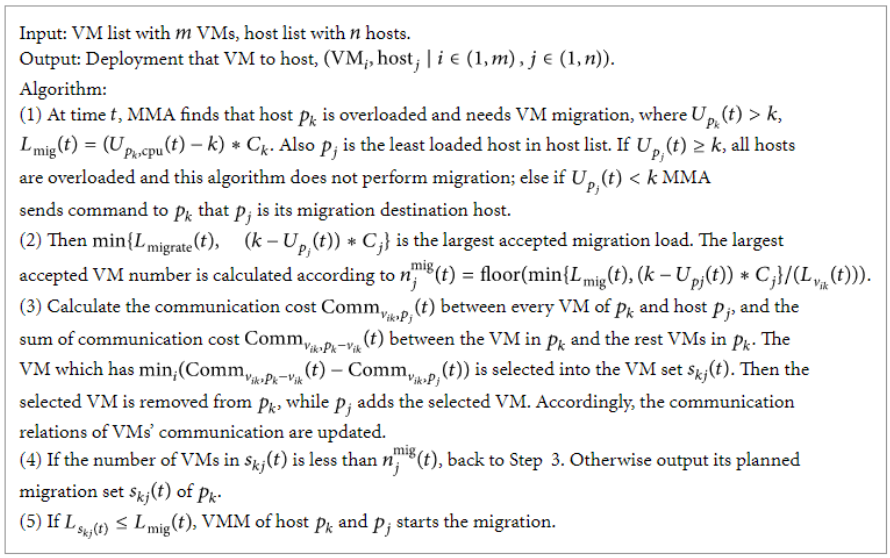
Where U_{cpu} *(t)* is CPU utilisation of host P_j (in percent) and U_{mem} *(t)* is memory utilisation of host P_j (in percent). α is a coefficient representing the relative importance between CPU utilisation and memory utilisation. As both CPU and memory are equally important for running VM, α is set to 0.5. For the determination of utilisation threshold, the following equation is used: $k = [\,(1+\beta)\,((\sum_{j=1} U_{pj})/n)]$, where k is the CPU utilisation threshold for all hosts U_{pj} is the load of node j, n is the number of hosts in the simulation system, and $0 \leq \beta \leq 0.2$ is a normalised constant. To generate migration $\beta = 0.1$ is set.

There are two load states for the hosts

1. If $U_{pj}\,(t) \geq$ k, host P_j overloaded
2. If $U_{pj}\,(t) <$ k, host P_j underloaded

The Migration Management Agent algorithm applies live migration to migrate VMs from an overloaded host to the least loaded host and ensures the physical machines would not be

overloaded and crash. Figure 4.4 shows the communication cost based VM dynamic migration algorithm.



Input: VM list with m VMs, host list with n hosts.
Output: Deployment that VM to host, $(\mathrm{VM}_i, \mathrm{host}_j \mid i \in (1,m), j \in (1,n))$.
Algorithm:
(1) At time t, MMA finds that host p_k is overloaded and needs VM migration, where $U_{p_k}(t) > k$, $L_{\mathrm{mig}}(t) = (U_{p_k,\mathrm{cpu}}(t) - k) * C_k$. Also p_j is the least loaded host in host list. If $U_{p_j}(t) \geq k$, all hosts are overloaded and this algorithm does not perform migration; else if $U_{p_j}(t) < k$ MMA sends command to p_k that p_j is its migration destination host.
(2) Then $\min\{L_{\mathrm{migrate}}(t), \quad (k - U_{p_j}(t)) * C_j\}$ is the largest accepted migration load. The largest accepted VM number is calculated according to $n_j^{\mathrm{mig}}(t) = \mathrm{floor}(\min\{L_{\mathrm{mig}}(t), (k - U_{pj}(t)) * C_j\}/(L_{v_{ik}}(t)))$.
(3) Calculate the communication cost $\mathrm{Comm}_{v_{ik},p_j}(t)$ between every VM of p_k and host p_j, and the sum of communication cost $\mathrm{Comm}_{v_{ik},p_k - v_{ik}}(t)$ between the VM in p_k and the rest VMs in p_k. The VM which has $\min_i(\mathrm{Comm}_{v_{ik},p_k - v_{ik}}(t) - \mathrm{Comm}_{v_{ik},p_j}(t))$ is selected into the VM set $s_{kj}(t)$. Then the selected VM is removed from p_k, while p_j adds the selected VM. Accordingly, the communication relations of VMs' communication are updated.
(4) If the number of VMs in $s_{kj}(t)$ is less than $n_j^{\mathrm{mig}}(t)$, back to Step 3. Otherwise output its planned migration set $s_{kj}(t)$ of p_k.
(5) If $L_{s_{kj}(t)} \leq L_{\mathrm{mig}}(t)$, VMM of host p_k and p_j starts the migration.

Figure 4.4: Communication Cost Based VM Dynamic Migration Algorithm (Song et al. 2015)

The above algorithm (shown in figure 4.4) calculates the communication cost between VMs and host and select the migration path with the least communication cost. Based on the simulation and realistic platform results, it is observed that the number of overloaded hosts is reduced.

4.7.2 Adaptive Distributed Load Balancing Algorithm Based on Live VM migration

Zhao et al. presented a distributed load balancing algorithm that is based on comparison and balance (DLBA-CAB) by adaptive live VM migration. This algorithm has been implemented to address lack of load balancing in EUCALYPTUS open source cloud computing framework (Zhao et al.2009). A program called Iblog, which is a Cron job (a Cron is a time-based job scheduler which runs on the server) that runs on each physical server to monitor predefined resources such as CPU usage and I/O load. Collected resource utilisation data record on the shared storage, then the distributed load balancing algorithm (DLBA) runs on each physical

server separately and dynamically migrates virtual machines from local physical machines to other physical hosts according to the resources usage (Zaho et al. 2009).

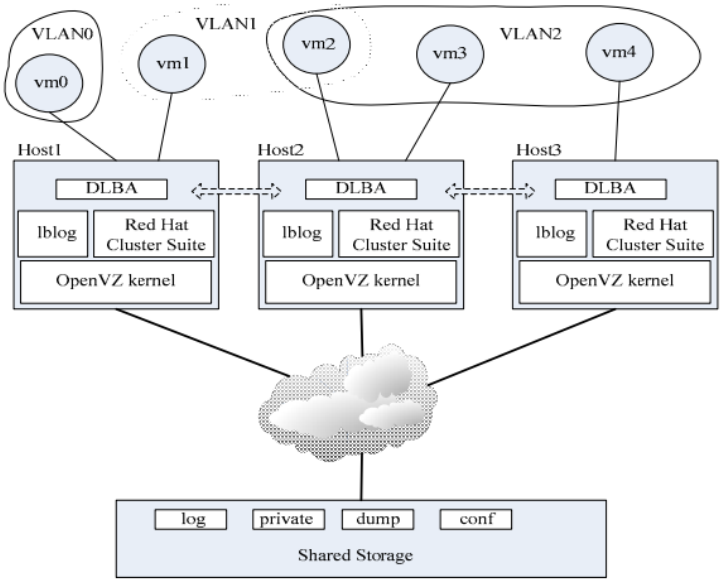


Figure 4.5: EUCALYPTUS System Architecture (Zaho et al. 2009)

The Compare and Balance Load Balancing Algorithm in EUCALYPTUS Framework (figure 4.5) is intended to be used in large distributed systems, and it runs on each physical server concurrently, but they don't communicate with each other. In order to get the total number of virtual machines that are running in the system, it uses Red Hat Cluster suit instead of querying all physical hosts. Its objective is to make each host to achieve equilibrium CPU usage and I/O usage.

Algorithm COMPARE_AND_BALANCE

for each VM i on current host cur do
  for all host $j \in [m]$ do
    $n_j \leftarrow$ num of VMs on host j
  end for
  $n \leftarrow \sum_{j \in [m]} n_j$
  choose $k \in [m]$ randomly with $P[k] = n_k / n$
  $c' \leftarrow$ MEASURE_COST(k)
  $c \leftarrow$ MEASURE_COST(cur)
  if $c' < c$ then
    with probability $c - c'$ migrate to k
  end if
end for

Figure 4.6: Compare and Balance Algorithm in EUCALYPTUS System (Zaho et al.2009)

Authors modelled a cost function considering CPU usage and I/O usage, and each host calculates the function values individually.

In each monitor intervals, 2 hosts are selected randomly and calculate to find the cost difference between CPU usage and the I/O usage. The difference is regarded as migration probability, in which the VMs are always migrated from the physical hosts with a higher cost to those with a lower one. After completing migration, the algorithm enables the system to reach a Nash equilibrium that reflects the loads are well balanced.

This system takes only CPU usage and I/O usage into consideration when triggering VM migration and it doesn't consider memory use or available bandwidth in the network. Authors assumed that each physical host has enough memory when migrating VMs from local host to other hosts.

4.7.3 Central Load Balancing Policy for Live VM migration

Bhadani eta al proposed a central load balancing policy for VM migration that is based on distributed environment to archive shorter response time and higher throughput (Bhadani et al. 2010). In that system, it denotes each guest as a job and each server as a node. The authors proposed a load balancing policy called, "Central Load Balancing Policy for Virtual Machines" (CLBVM), which takes load balancing decisions based on global state

information. The CLBVM policy takes into account the following considerations (Bhadani eta al. 2010).

1. Network load is constant and does not change frequently.
2. Each VM has different identifications.
3. The load information collector daemon process runs continuously in each pServer and it collects CPU load and system utilisation by guests. Based on the collected data, it categorises pServers as Heavy (H), Moderate (M), or Light (L).
4. The messages are exchanged with the Central server which takes decisions periodically for load balancing. Heavily loaded machines are balanced by periodically reallocating the loads on heavily loaded servers to the lightly loaded servers. If all servers are evenly loaded, VM migration will not be performed.
5. Frequent change in state is taken into consideration by the load balancing algorithm, such that prevents unnecessary VM migrations. Figure 4.7 shows the experimental setup of the CLBVM system.

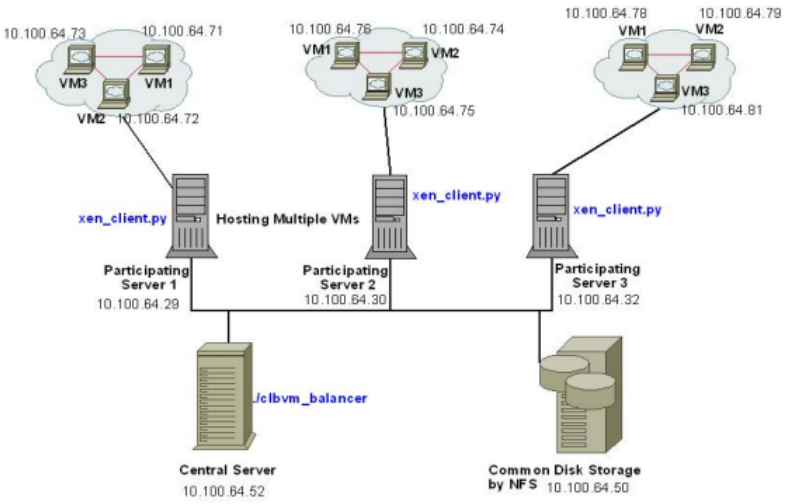


Figure 4.7: The experimental setup of CLBVM system (Bhadani et al. 2010)

This system has some drawbacks, it assumes that the network loads are almost constant, but it's not applicable in the current Cloud environment that changes frequently. Other limitations are the resource types such as memory and I/O which are rarely considered in this work.

4.7.4 Optimised Control Strategy Combining Multistrategy and Prediction Mechanism

Yang et al proposed a Multistrategy prediction mechanism for VM migration to reduce the number of overloaded hosts and avoid unnecessary migration (Yang et al.2011). The authors also adopted a weighted function that considers multiple types of resources.

To understand the load of the physical hosts, the authors defined four status domains, light-load domain, optimal domain, warning domain, and overloaded domain.

Light-load domain –The host has a light load

Optimal domain – The host lies on the optimal status.

Warning domain – This is a buffer domain between the optimal domain and the overload domain. At this stage, a suitable strategy must be adopted to avoid the host being overloaded.

Overloaded domain – The host is overloaded

To predict future utilisation of hosts, the system has a prediction model that uses a set of recently used data series based on an autoregressive (AR) prediction model. When migrating VMs, it considers the characteristics of applications such as CPU utilisation and I/O utilisation. The destination host for migrating VMs will be selected by looking at the resources' fluctuation of hosts, such as CPU utilisation. Since that is the most suitable and influential one, the host with the largest CPU resource is selected as the destination. The four variables above present the trend degree about the utilisation change of CPU, memory and I/O and network bandwidth.

(1) If the CPU utilisation change trend is the leading contributor for overloading the host then VM migration policy selects a VM or VMs to migrate that is based on the following steps:

Step1: According to the recent CPU utilisation of VMs that are running on the host, the strategy picks VMs that have the highest CPU utilisation, those VMs are called the Candidate set (candidate VMs for migration).

Step2: We define the variable $VML_i = \frac{\text{CPU Utilisation}}{\text{Memory Utilisation}}$

According to this definition, the higher the CPU utilisation or the smaller the memory utilisation is, the greater the value of VML_i will be. The VM migration strategy will select the VMs that have the highest CPU utilisation trend and least memory utilisation.

(2) If the memory utilisation change trend is the greatest, migration strategy will select the VMs that occupy the most memory.

(3) If the I/O change trend is the greatest, migration strategy will select the VMs for migration according to following steps:

Step1: Select the VMs with the rising I/O utilisation trend, those VMs which are called the Candidate set.

Step 2: Define the variable VML_i for each of the Candidate set,

$$VML_i = \frac{I/O\ \ Utilisation}{Memory\ Utilisation}$$

The VM strategy selects the VMs with higher I/O utilisation or less memory utilisation, the value VML_i will be greater.

(1) If the network bandwidth utilisation change trend is the biggest, the migration strategy will select a VM according to the following steps:

Step i: Migration strategy will select a VM with the highest bandwidth utilisation trend, and those set of VMs are called the Candidate set.

Step ii: Authors define the variable VML_i, for each Candidate set as follows:

$$VML_i = \frac{Network\ Utilisation}{Memory\ Utilisation}$$

When VM migration policy selects the VMs with the highest bandwidth utilisation trend or smaller memory utilisation, the greater value of VML_i will be. Flow charts for each four status domains are shown in the figures 4.8, 4.9 and 4.10.

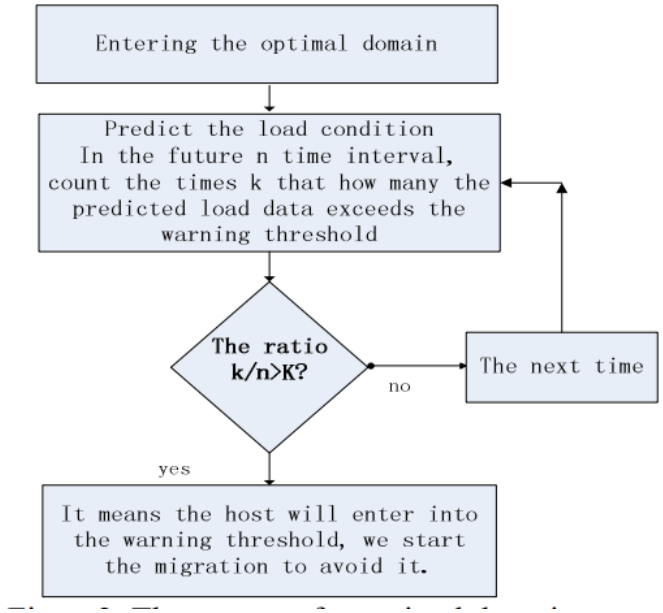


Figure 4.8: The Strategy for Optimal Domain (Yang et al. 2011)

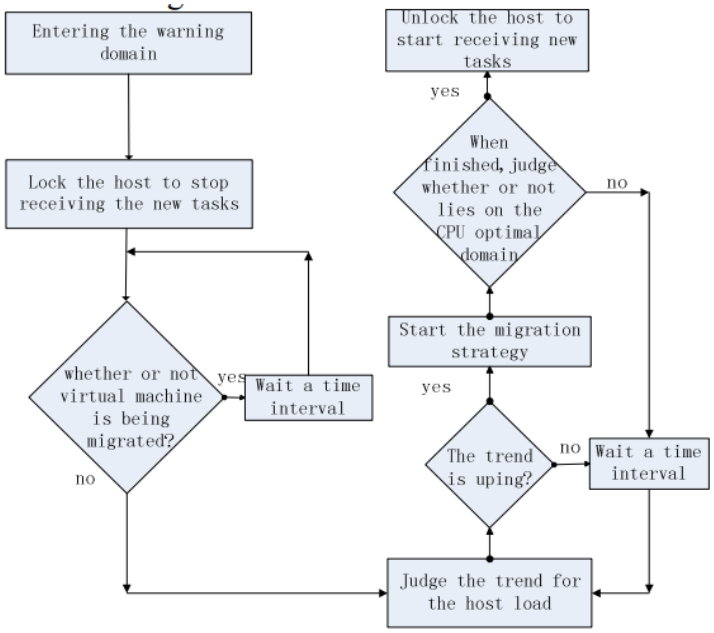


Figure 4.9: Strategy for Warning Domain (Yang et al. 2011)

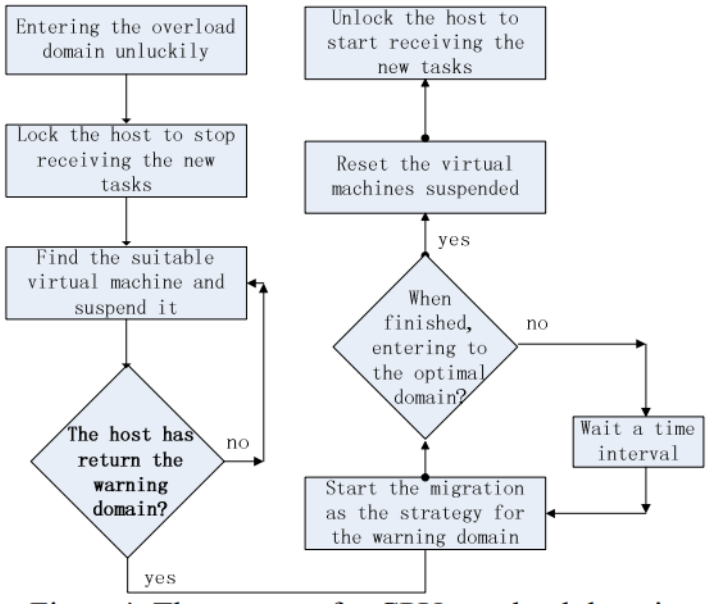


Figure 4.10: Strategy for CPU Overload Domain (Yang et al. 2011)

Migration Destination Choice

In this system, a 3 times handshaking protocol (Figure 4.11) is used to find the best host for migrating VMs, the most important aspect of this protocol is to avoid multiple VMs migrating to the same host and overloading simultaneously. In order to avoid this situation, the destination should increase the load value once it agrees request for migration from the requester (Yang et al. 2011). The destination should use the increased load to decide whether or not to agree the next request. The figure 4.11 shows the process of the three time handshaking protocol.

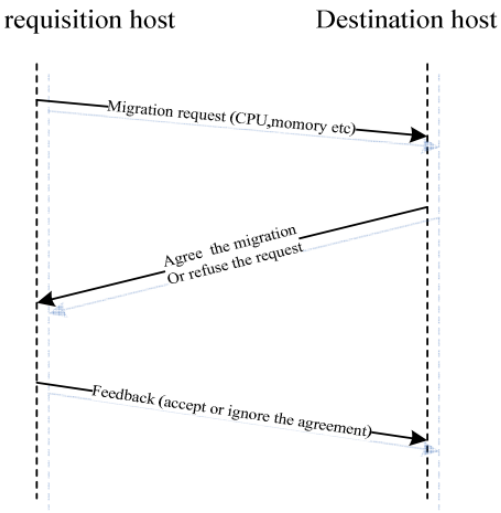


Figure 4.11: The Three Time Handshaking Protocol (Yang et al. 2011)

With this protocol, each host maintains an acceptance queue containing VMs that are waited for allocation, and this queue updates host utilisation and load increment along with time.

The advantage of this algorithm is, it has the ability to adapt into different situations as it has four different strategies for different host status. However this algorithm is suitable for small scale data canter scenarios but not for modern cloud data centres which are equipped with thousands of VMs.

4.7.5 Hybrid Genetic-Based Host Load Aware Algorithm

Thiruvenkadam et al proposed a hybrid genetic based host load-aware algorithm for scheduling and optimising VMs. The authors' main objective is minimising the number of VM migrations while balancing the load of physical machines. They focused on variable loads and dynamicity of VM allocations, therefore their framework architecture has two functions, one is checking the loads of host and user constraints and other is optimising VM placement by applying the proposed algorithm called a multidimensional physical host load-aware scheduling and hybrid genetic based optimisation algorithm (Thiruvenkadam et al.2015).

This method identifies the overload physical machines by taking the physical machine resources' (CPU, memory and network bandwidth) utilisation into consideration. The centralised controller of the system stores hosts historical and current loads globally. It then selects VM to migrate based on the past behaviour of VMs and picks the appropriate PM based on its resource utilisation rate. It also discovers underutilised servers and migrates the VMs running on it to some other suitable servers to save energy.

Where $PM = \{ PM_1, PM_2, PM_3, \ldots PM_m \}$ and m is total number of physical machines while an individual physical machine can be denoted as $PM\ i$, where i denotes the physical machine number and range of i .Similarly, the set of VMs on the physical machine i can be $\{ VMi1, VMi2, \ldots . VMin \}$ here n is the number of VMs on the physical server i. The CPU , RAM and bandwidth usage have to be calculated individually in order to deploy VMj on the PMi . The CPU load of all VMs of PMi at the time of interval ts can be calculated using the equation 4.1.

$$PMi(cpu, ts) = \sum_{j=1}^{n} VMij\ (cpu, ts) \qquad Eq\ (4.1)$$

The amount of RAM usage by all VMs of PMi at the time of interval ts can be calculated using equation 4.2.

$$PMi(ram, ts) = \sum_{j=1}^{n} VMij\ (ram, ts) \qquad Eq\ (4.2)$$

The amount of network bandwidth usage by all VMs of PMi at the time of interval ts can be calculated using equation 4.3.

$$PMi(nbw, ts) = \sum_{j=1}^{n} VMij\ (nbw, ts) \qquad Eq\ (4.3)$$

$$PMCk(WL, ts) = \sum_{j=1}^{m} PMi\ (WL, ts) \qquad Eq\ (4.4)$$

Where $PMCk$ represents the k^{th} physical machine cluster of the datacentre, WL represents the weighted load of the PM cluster k. Where PMi represents the i^{th} physical machine of the PM cluster k, VMij represents j^{th} virtual machine of the, and m is total number of physical machines in the cluster *k*. The weighted average load of the physical machine cluster k at any time interval ts can be denoted using the equation 4.4.

The weighted load of k^{th} physical machine cluster of the datacentre should not be exceeded the host capacity, which can be denoted in the equation 4.5.

$$\sum PMi\ W_{resource\ usage}\ (ts) \leq TH\ value \leq \sum PMi\ W_{resource\ capacity} \qquad Eq\ (4.5)$$

Where resources {CPU, RAM, Network Bandwidth} and $W_{resource}$ are the weight associated with each resource, *TH* value is the threshold value that is set by the administrator. If the load goes beyond the *TH* value, the host can be considered as overloaded host (Thiruvenkadam et al. 2015). For VM selection, authors proposed a heuristic approach based on multiple policies, by searching PM machines according to the VM resources utilisation. The genetic algorithm keeps running and searching optimised solutions until the metrics are satisfied (Thiruvenkadam et al. 2015).

The problem with this meta-heuristic approach is that it requires more time to process than heuristic algorithms, and also it increases the implementation complexity in a realistic cloud environment.

4.7.6 Prepartition Algorithm for VM Load Balancing in Cloud

Tian et al proposed an algorithm for off-line VM allocation within the reservation model, which is called Prepartition. In this method, all VM requests are reserved and all VM information is gathered before the final placement. In the reservation model, the VM requests are partitioned into smaller ones which enable more efficient utilisation of resources and reduce overload machines. Virtual machines with multiple resources are considered in this method, and also authors redefined the traditional makespan as a new metric capacity makespan, which is computed as VM CPU load multiplies VM capacity. The VMs are allocated one by one to the host with the lowest makespan capacity (Tian et al. 2014).

As mentioned in the literature review, the traditional makespan measures the total processing time of the load balancing algorithms. The authors' goal is to minimise maximum load (capacity_makespan) on any PM (Tian et al. 2014). The capacity makespan of any PM can be calculated using the below equation (eq4. 6).

$$CM_i = \sum j \in A(i)\ d_j t_j \qquad Eq\ (4.6)$$

Capacity makespan of PM: Where in any allocation of VM requests to PMs, let $A(i)$ denote set of VM requests that are allocated to the physical machine PM_i. Under this allocation, machine PM_i will have total load equal to the sum of product of each required capacity and its duration. Where d_j is the capacity request of VM_i from a PM and t_j is the span of request j (the length of processing time of request j).

Simulated with heterogeneous cloud and real traces, the authors illustrated the proposed algorithm. The prepartition algorithm achieved lower average makespan and capacity makespan, the pseudo code of preparation algorithm is shown in figure 4.12.

Input: VM requests indicated by their (required VM type IDs, start times, ending times, requested capacity), CM_i is the capacity_makespan of request *i*.

Output: Assign a PM ID to all requests and their partitions (migrations)
Initialization: computing the bound P_0 value, set the partition value *k*.

1: IF $CM_i > (P_0/k)$, *THEN partitions it into multiple* P_0/k *subintervals equally* and consider each subinterval as a new request
2: Sorts all intervals in decreasing order of *CMs*, break ties arbitrarily;
3: Let $I_1, I_2, \ldots, I_n$ denote the intervals in this order
3: **For** *j* from *1* to *n* ***Do***
4: allocates *j* to the PM with the lowest load and available capacity
5: updates load (*CM*) of the PM
6: **Endfor**
7: Computes *CM* of each PM and total partitions (migrations)

Figure 4.12: The Pseudo Code of the Prepartition Algorithm (Tian et al. 2014)

4.7.7 Network-Aware Migration Control and Scheduling Differentiated VM Workloads

Thomas Stage et al. had proposed a framework called Network-Aware Migration Control and Scheduling Differentiated Virtual Machine Workloads (Stage et al.2009). It underlines the necessity for additional control parameters for efficient migration scheduling. According to Alexander Stage et al. multiple consolidated workloads on a physical host require a corresponding network capacity (Stage et al.2009). Network topologies in data centre are designed as multi rooted trees, depicted in figure 4.13, where higher workload in combination with an intensive network bandwidth may lead to network contention. They introduced a network aware virtual machine migration method to ease the load of physical hosts and proposed a scheme for classifying VMs based on their workload and taking networking bandwidth requirements of VM migration and network topologies.

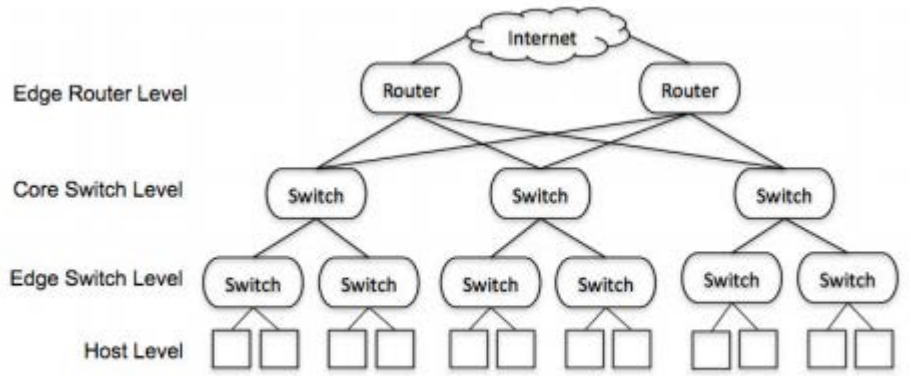


Figure 4.13: Network Topology in a data centre (Stage et al.2009)

There are four main components in the proposed architecture, Workload Classifier, Allocation Planner and Migration Scheduler. Figure 4.13 shows the network topology in a data centre.

- Workload Classifier - Classifies workload according to three main workload attributes which are: predictability, trend, and periodically.
- Allocation Planner - Based on classified workloads, the allocation planner predicts underloaded or overloaded physical machine situations.
- Non- Conformance detector – Handles unexpected sudden surge of resources demands.
- Live Migration Scheduler – Is responsible for scheduling VM migration in order to avoid VM migration related Sla violation.

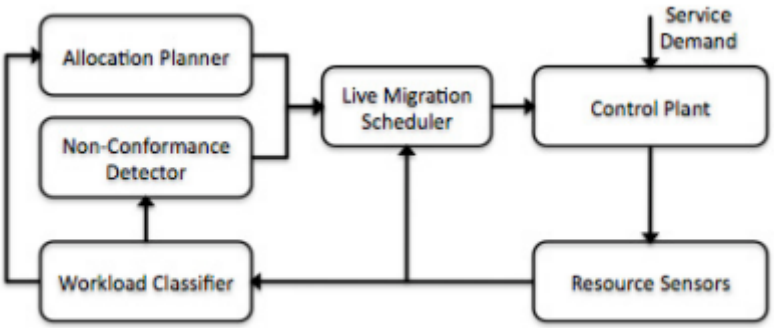


Figure 4.14: Control System Architecture (Stage et al.2009)

There are three main components in the architecture which is shown in figure 4.14, Workload Classifier, Allocation Planner and Migration Scheduler.

For efficient VM migration scheduling, bandwidth adoption behaviour including minimum and maximum bandwidth usage and iteration termination conditions need to be controlled. There are three main components in the proposed Network Aware Migration Control architecture: workload classifier, allocation planner and migration scheduler. In the workload classifier, physical hosts' workloads are classified and grouped based on historical data which has been gathered from different clusters in the network. In this method they used static VM consolidation; the method concerned is suitable for small data centres where user demand is predictable but not suitable for large data centres where demand is unpredictable. In this architecture, they ignored CPU and memory usage statistics of physical hosts. By looking at the physical host's CPU and memory usage, it can predict overloaded and underloaded physical machines dynamically.

In addition, the proposed architecture ignored bandwidth completion between user traffic and VM migration traffic. When a VM was migrating from a source host to a destination host, the user traffic bound to the destination host competes for bandwidth with the VM migration traffic. To secure bandwidth for VM migration, it is necessary to control the bandwidth of user traffic.

4.7.8 Autonomous agent to Schedule Live VM migration

Martin Duggam et al. proposed an autonomous agent which was developed from an artificial intelligence technique known as Reinforcement Learning for dynamic selection of VM in Cloud (Duggam et al. 2016). It acts like a decision support system that enables an agent to learn an optimal time to schedule a virtual machine migration depending on the current network traffic demand. According to the authors, data centre network traffic varies according to the time during a week, a day, an hour and a minute. The time is an important factor when considering VM migration, because various network traffic demands could increase or decrease the total VM migration time. The main aspect of this proposed method is to avoid needless VM migration due to just a small spike of host workloads.

RL network aware VM migration strategy will have two representatives for its state space. The first state space S_t defined as current host CPU utilisation, denoted as h_u is returned as a

percentage. The first space can be defined as $S_t \in S = 0,100$ and it is obtained through the following equation (Eq4.7)

$$S_{t=\sum_{v=1}^{n} vmu}(V_n) \qquad (Eq4.7)$$

V_n is the current VM selected, vmu is the function that calculates the V_n utilisation of the host's CPU and n is the total of all possible VMs that can be migrated.

The second state space d_t is defined as the current direction of the network traffic. There are three categories in network traffic, *increase, decrease* and *level*. When d_t = *level*, there's no change in the network traffic, if d_t = *increase* this indicates the network traffic demand is increased. d_t = *decrease* indicates network traffic decreased. The action a_t space represents wait or migrate, when the RL agent selects wait then VM migration will not happen. But when the agent decides to select migrate then the scheduled group of VM migration will occur.

Researchers who proposed this method were trying to avoid threshold based VM migration by implementing an artificial intelligent agent which is based on Reinforcement Learning. In this method, they only consider host CPU utilization and bandwidth in VM migration, but they ignored user traffic demands for VMs. Virtual Machine's memory utilization and user traffic demands must be taken into consideration in VM migration, because moving too much memory would increase the VM migration time.

4.7.9 VM placement method based on network bandwidth

Konomi Mochizuki et al. proposed a VM placement method based on network bandwidth. According to Mochizuki et al. they are investigating the edge cloud architecture to achieve the flexible management and operation of the edge cloud's (Figure 4.15) functions in a network. Developing an edge cloud means that edge functions need to be virtualised affectively, a virtualised edge function can be archived by implementing the edge function on a virtual machine and the VM migration technique can be used to relocate the VMs without disrupting services. They proposed a method to control bandwidth and determine the order of VM migration required to complete all VM migrations while ensuring the minimum bandwidth of user traffic (Mochizuki et al. 2013).

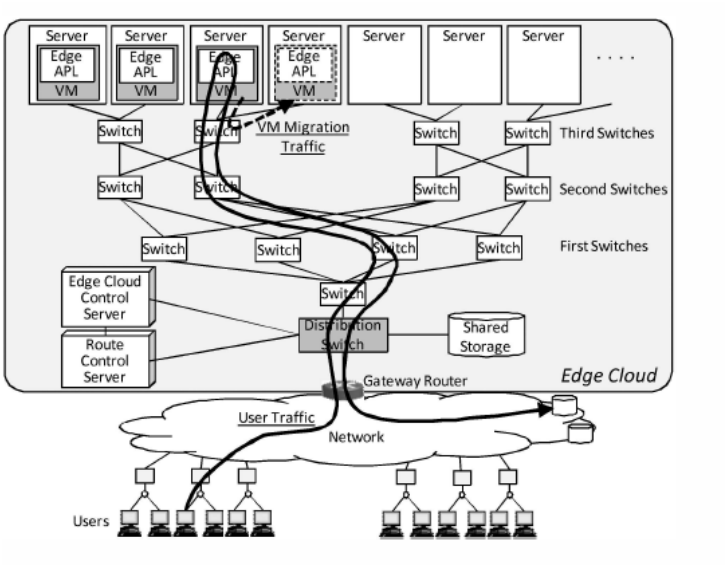


Figure 4.15: Example of network edge cloud (Mochizuki et al. 2013)

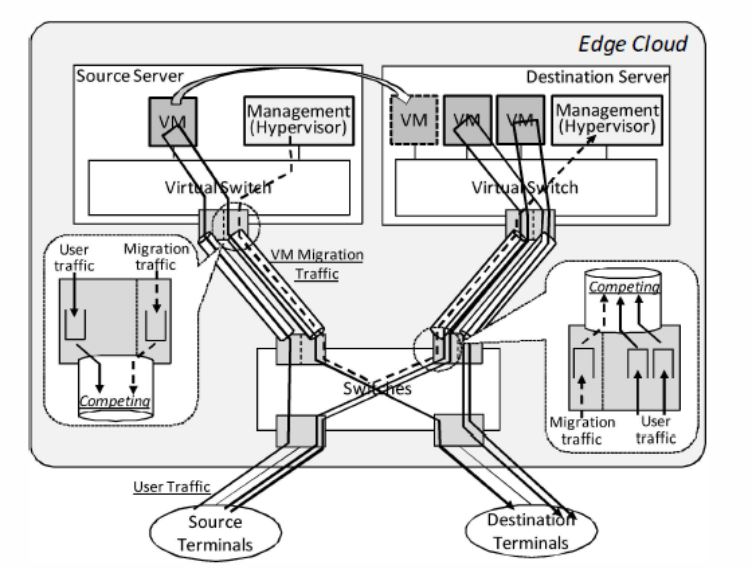


Figure 4.16: Competition for bandwidth by user traffic and VM migration traffic (Mochizuki et al. 2013)

When VM occurs, it moves VM memory from the source physical machine to the destination physical machine through the network. The migrating VM memory size is M while the amount of VM state change per second is set to W, and the time migration takes is set to t. The total amount of information $M + W_t$ will be transmitted to the destination server in order to transfer information $M + W_t$ over the link bandwidth B_m and complete the migration in time t, (eq 4.8)

$$M + W_t = B_m t \qquad (Eq4.8)$$

By transforming eq 4.8, the migration time *t* can be obtained using the equation 4.9 below:

$$t = \frac{M}{B_m - W} \qquad (Eq4.9)$$

When relocating a VM or a few VMs together, the number of VMs on the source server or the destination server will change at every migration of a VM. The state of the bandwidth competition (figure 4.16) differs due to every VM migration and also the link bandwidth B_m . The amount of traffic processed by the VMs would also vary according to the workload of the VM. As a result *W* likewise varies from *VM* to *VM* and because of that, virtual machines which have a large *W* value will take a long time to migrate from source to destination. To solve this problem, authors of this paper proposed a method to migrate VM that has suitable W values according to the changing B_m. The order of migration is set sequentially starting with the VM that requires the largest bandwidth for migration (the highest W_i value) and the VM with the smallest W_i value will be migrated at the end of the relocation.

In this research authors assumed that one VM operates on each physical server, and there is a mutual connectivity between servers through switches. Each server has routes with the same bandwidth in both directions. And also they assumed that the same application is running on each VM, and the amount of VM memory is equal *(M).* In this research, authors proposed a method to control bandwidth and a method to determine the order of the migration while ensuring the minimum guaranteed bandwidth of user traffic. The order of the VM migration is determined by the amount of bandwidth needed to migrate a VM from host to destination, and it is assumed every VM has a similar memory size. But a cloud data centre equipped with heterogeneous VMs need a dynamic method to understand the actual memory size of the migrating VMs. The total migration time of a VM depends on the size of the VM and the amount of bandwidth on the network, applying a VM memory of a static size is not suitable for a dynamic environment like cloud. And also this method lacks technique to detect overloaded and underloaded machines, plus VM selection and a VM placement method.

4.7.10 Black Box and Grey Box Framework for VM Live Migration

Wood et al. proposed an automated VM live migration framework called Black box and Gray box for VM migration in large data centres. In this strategy, it automates the tasks of monitoring resource usage, hotspot detection; determining a new mapping and initiating the necessary migration (Wood et al. 2007) by monitoring VM resources. Later they identified limitations of the black box and gray box approach and proposed an algorithm called

"Sandpiper" (Figure 4.17) which is a hotspot that determines when to migrate VMs, how and where to.

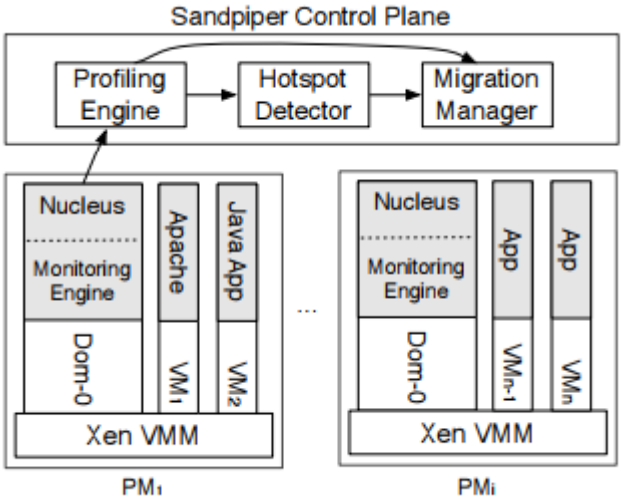


Figure 4.17: The Sandpiper Architecture (Wood et al.2007)

The black box monitoring engine resides in every VM and tracks CPU, network and memory usage of each VM. It also calculates total resources usage on each PM by aggregating the usages of resident VMs. CPU monitoring carried out by observing the number of network I/O requests by each VM. Network monitoring is carried out by observing the number of bytes sent to each virtual network interface. Memory usage is inferred from the number of read/write requests to swap partitions.

Sandpiper runs a component called the "nucleus" on each physical server; the nucleus runs on a special virtual server and it is responsible for gathering resources' usage statistics on the corresponding physical server. It gathers CPU, network bandwidth and memory usage of the physical server.

Gray box monitoring is useful in scenarios where it is not feasible to "peek inside" a VM to gather usage statistics. It gathers OS level statistics as well as application logs which potentially enhance the quality of decision making in Sandpiper.

Hotspot Detection

The hotspot detection algorithm is responsible for triggering VM migration whenever SLA violations are detected by black box and gray box approaches. Hotspot detection is performed on a per-physical server basis in the black box approach, and hotspot is flagged if the

aggregate CPU or network bandwidth on the physical server exceeds a threshold. SLA violation must be detected on a per-virtual VM basis in the gray box approach. In the gray box approach, it flags a hotspot if the memory utilisation of the VM exceeds a threshold or if the response time or request drop rate exceed the SLA specified values (Wood et al. 2016).

The black box and gray box framework is a centralised load balancing system that is implemented to automate the tasks of the monitoring system resource usage of VMs, hotspot detection, VM selection, and initiation of the necessary migrations. The sandpiper control plane runs on one of the servers in the cluster and contains three components; a profiling engine, a hotspot detector, and a migration manager.

The profile engine uses the statistics that are sent from the nucleus to construct a resources usage profile for each VM and aggregate profiles for each PM. The hotspot detector continuously monitors these usage profiles to detect the hotspot, which is said to have occurred if the aggregate usage of any resources (processor, network, or memory) exceeds a threshold or if SLA violations occur for a “sustained” period.

The VM migration occurs according to the following algorithm:

1 Physical servers are sorted in a decreasing order of Volume (Load)

1. VMs are sorted in a decreasing order of Volume to Size ratio (VSR), size being the memory foot print of the VM.
2. Attempt to fit the VM with the highest VSR (highest memory) in the Physical Server with the highest volume (load) into a physical server with the lowest volume.
3. If this is not possible, the Physical Server with the second lowest Volume is considered. This continues until a match is found.
4. If a match is not found, the VM with the second highest VSR is tried to enable movement in a similar fashion. This process continues until the Physical Server’s resources utilisation falls under the threshold.
5. Now the process is repeated for the Physical Server with the second highest Volume and so on.

The above implementation covers PM overload detection, VM selection, VM migration and VM resource provisioning. According to the algorithm it first selects VMs that have the highest volume and PMs that have the least volume. The following equation is used to calculate the VM and PM volumes.

Vol = CPU Load × Network Load × Memory Load

Consider a PM having VM_1, VM_2, and VM_3 with utilisation as follows;

	CPU Utilisation	Network Utilisation	Memory Utilisation
VM1	80 %	50%	10%
VM2	80%	70%	10%
VM3	15%	80%	80%

Table 4.1*:* VMs Resources Utilisation in the Back Box and Grey Box Framework

We assume that the threshold value is 75 %, according to the Sandpiper algorithm the hotspot will be detected for VM1 and VM2. But it will select the VM that has the highest volume, so VM_3 will be migrated. But moving VM_3 no significant improvement will be made since it uses only 15% of CPU. In this approach, when moving a VM, it temporarily moves to the Control Pane of the framework and runs until it's migrated to the destination host, but the authors failed to mention who monitors the CPU, network and memory utilisation during this temporary period. In addition the Sandpiper algorithm doesn't consider the network bandwidth when moving VMs over the network, and there is no mechanism to shorten the total migration time.

4.10.11 Heuristics Dynamic Load Balancing Approaches

Heuristics approaches for dynamic load balancing have been researched in this section; the Heuristics dynamic load balancing approaches apply a statistical analysis of the observed history of system behaviour to understand the status of the current system (Beloglazov. 2013). Heuristics means "act of discovering ", the heuristic algorithms are performed based on previous experience and knowledge gathered about the problem. Heuristic algorithms are tailored to address a specific problem rather than a general class of problems.

These approaches are mainly used for consolidation of VMs and minimising energy in cloud data centres, but the VM consolidation and minimisation of energy consumption strategies are out of the scope of this research; however some of the approaches can be applied when developing algorithms in the proposed framework.

4.10.11.1 Host Overload Detection

Each host periodically executes an overload detection algorithm to de-consolidate VMs in order to avoid performance degradation and avoid SLA violations. This section describes several heuristics proposed for the host overload detection problem.

A Static CPU Utilisation Threshold

This approach is one of the simplest overload detection strategies, the idea is to set up a CPU utilisation threshold distinguishing the overloaded and underloaded machines. When the algorithm invokes, it compares the current CPU utilisation of the host with a defined threshold. If the threshold is exceeded, the algorithm detects an overloaded host. Rajyashree et al. proposed a double threshold based load balancing approach by using VM live migration. The main objective of their research is reducing the number of migrations. The four steps that involved in the VM migration are stated below (Rajyashree et al.2015).

1. Calculate the load on the PM and VM
2. Calculate the upper and lower thresholds to find the overloaded and underloaded machines.
3. Selecting the best VM to migrate from the PMs.
4. Select the best host for migrating VMs.

The double thresholds are calculated based on the host CPU and RAM utilisation. If the load of the host is greater than the upper threshold, then the host considers it as overloaded. If the load is below the upper threshold, then the all VMs that are running on that host will move to another host (Rajyashree et al.2015).

An Adaptive Utilisation Threshold: Median Absolute Deviation

According to Anton, heuristic approaches for detecting host overloads based on simple static CPU utilisation threshold are unsuitable for an environment with dynamic and unpredictable workloads (Beloglazov et al.2012). He claims the system should be able to automatically adjust its behaviour depending on the workload patterns of applications He proposed a heuristics algorithm for auto-adjustment of utilisation threshold, which is based on statistical analysis of historical data collected from VMs. The proposed algorithm adjusts the value of the CPU utilisation threshold depending on the strength of the deviation of the CPU utilisation (Beloglazov et al. 2012).

4.10.11.2 VM Selection Policies for VM Live Migration

Once overloaded hosts have been detected, it is necessary to determine which VMs are the best to be migrated from the host in order to avoid performance degradation. Baghshahi et al. proposed a method to migrate VM from one data centre to another based on the Greedy Algorithm (Baghshahi et al. 2014). According to the authors, most VM migration techniques are considered by moving one VM or couple of VM at a time. But in this proposed method, it migrates the total VMs in a cluster using the Greedy algorithm. The Greedy algorithm starts with an empty set and adds items to the set in sequence until the set presents a solution to the problem. In this method, it takes the memory size of the virtual machine and the storage size of the destination physical machine into consideration. At each step, a comparison should be made between memory size of the VM and the storage size of the physical machine. The Pseudo code of the greedy algorithm is shown in figure 4.18.

```
1.  input:
2.  Hostlist ,Vmlist //Sorted Desc
3.  Curent_Time
4.  Link_Speed
5.  VmMigration_Time
6.  VmMigrationList_Time
7.
8.  For i:0 to Hostlist
9.      host: Host_LargSize in Hostlist
10.     while host>0
11.         vm: VM_ LargSize in Vmlist
12.             for j:1 to Vmlist
13.                 If vm>host then
14.                     vm: vm++ in vmlist
15.                 else
16.                     host:host - vm (size)
17.                     vm is in Migration
18.
        VmMigrationList_Time:Curent_Time +
19. (vm/Link_Speed)
20.                 vm:vm++ in Vmlist
21.     host:host++ in Hostlist
```

Figure 4.18: Pseudo –code of the greedy algorithm for virtual machine migration (Baghshahi et al. 2014).

The steps of the greedy algorithm (Figure 4.18)

1. Select an item from the list of destination physical machine
2. Perform the following until the destination physical machine has enough storage space.
3. Select an item from the list of virtual machines.

Compare the size of the selected virtual machine with storage space of the destination physical machine. If the size of the virtual machine was greater, go back to step 1 and select the next virtual machine. Otherwise, the virtual machine will be migrated to the selected physical machine. (Somayeh et al. 2014).

In this method the authors ignored CPU utilisation, user traffic and the link bandwidth. To migrate group of VMs in a cluster, the network of the data centre needs to have adequate bandwidth to support VM migration, and also it needs a framework to provide enough bandwidth to the VM migration.

The Random Selection Policy

Dhingra et al. proposed a framework to optimise the VM allocation in order to reduce the energy consumption in cloud based data centres. According to the framework, Selection Choice (SC) policy randomly selects a VM to be migrated on the basis of a uniformly distributed discreet random variable $X \stackrel{d}{=} \text{ur}(0,|V_1|)$, whose values index a set of VMs V_j allocated to host *j* (Dhingra et al. 2014) .

The Minimum Migration Time Policy

The Minimum Migration time Policy (MMT) migrates a VM that requires a minimum time to migrate from the source to the destination host relatively to other VMs allocated to the host. The migration time estimated as the amount of RAM utilised by the VM divided by the spare bandwidth that is available for host *j*. Let V_j be a set of VMs currently allocated to the host *j*, the MMT policy finds a VM *v* that satisfies the following equation, (Beloglazov et al. 2012).

$$v \in V_j \mid \forall a \in V_j, \frac{RAM_u(v)}{NET_j} \leq \frac{RAM_u(a)}{NET_j} \qquad (Eq4.10)$$

$RAM_u(a)$ is the amount of RAM currently utilised by the VM a ; and NET_j is the spare bandwidth available for host *j* .

4.8 Chapter Summary

This chapter presented a comprehensive analysis of dynamic load balancing based on live VM migration with an analysis of load balancing algorithms and VM load balancing implementations. Live VM migration is a concept that is moving a VM from a source server to a destination host without interrupting its applications and users. Live VM migration can be used as a load balancer to distribute VMs dynamically across all physical nodes to balance workload, ensuring no single physical node is overwhelmed by excessive workload. Live VM migration can also be used to archive fault tolerance and conserve energy in data centers.

Load balancing algorithms can be divided into two main categories, which are static and dynamic, both types of load balancing algorithms have been discussed. Generally, load balancing algorithms in cloud can be implemented in two ways, as a Centralised system or as a Distributed system. In a centralised load balancing system, a central component or the scheduler obtains global information (utilisation, load, connections and other information) from hosts in the cluster. It then sends the information to the central scheduler which takes decisions according to the requirements of the system. Example of a commercial centralised load balancing system is the Red Hat Enterprise Virtualisation Suite. The disadvantage of a centralised system is a single point of failure, but it can be avoid using a backup server.

Distributed schedulers should be located on each node rather on a central node, and each node is responsible for making decisions in distributed way. By doing that, distributed algorithms eliminate bottlenecks and improves reliability and scalability, because it avoids a single point of failure.

The challenges of implementing load balancing algorithms in a cloud environment depend on following factors, network overhead, performance and reliability. Network overhead determines the amount of overhead involved while implementing load balancing. Overhead can be produced due to VM migration cost and communication cost of load balancing algorithms. Performance determines the efficiency of the load balancing system, and it includes following factors, resource utilisation, scalability and response time. Reliability is an important factor when implanting a load balancing system, it must avoid a single point of failure and provide minimum unplanned downtime.

The load balancing scheduling process is discussed in this chapter. Scheduling VM migration is divided into two main categories: VM initial placement stage and VM live migration stage.

In the VM initial placement stage, it doesn't consider live migration, instead focuses on selecting a suitable host for migrating VMs. Generally, host selection algorithms take the host's available resources into consideration.

There are different types of load balancing algorithm evaluation metrics, these metrics are optimised on the basis of different behaviours, such as obtaining maximal or minimal values. The following are some of the prominent adopted metrics in VM load balancing algorithms. These load balancing algorithm evaluation metrics have been discussed in the chapter.

- Load variance and standard deviation of utilisation
- Makespan
- Associated Overhead
- Throughput
- Number of Overloaded Hosts
- Average Imbalance Level
- Imbalance Score
- Number of Migrations

In this chapter a few existing VM load balancing implementations have been discussed and analysed. Song et al (Song et al.2015) proposed a VM migration management agent or algorithm to balance the load dynamically in High-Level Application (HLA) federations. The objectives of this approach are reducing the load of overloaded hosts and decreasing the communication cost among different federations. In this algorithm, the authors only considered host CPU utilisation for detecting overloaded and underloaded physical machines and VMs. However, CPU utilisation is not sufficient for detecting overloaded and underloaded machines in a large cloud data centre. Considering only CPU utilisation of the host will trigger unnecessary VM migration due to small spikes of the workload and it will cause performance degradation. Therefore multiple host resources' utilisation must be considered when designing underload detecting algorithms.

Zhao et al. presented a distributed load balancing algorithm that is based on comparison and balance (DLBA-CAB) by adaptive live VM migration. This algorithm has been implemented to address lack of load balancing in a EUCALYPTUS open source cloud computing framework. This framework is intended to be used in a large distributed system and the load balancing algorithm runs on each physical server concurrently but they don't communicate with each other. This framework doesn't have a mechanism to find the state of the VMs and

PMs in the system or their resource consumption. This system takes only CPU usage and I/O usage into consideration when triggering VM migration and it doesn't consider memory use or available bandwidth in the network. Likewise it doesn't consider the host physical machine's RAM when placing VMs on the host, because the VM cannot move to the host PM if the VM's RAM is larger than the host RAM. The authors assumed that each physical host has enough memory when migrating VMs from local host to other hosts.

Bhadani et al proposed a central load balancing policy for VM migration that is based on a distributed environment to archive a shorter response time and higher throughput. In that system, it denotes each guest as a job and each server as a node. The authors proposed a load balancing policy called, "Central Load Balancing Policy for Virtual Machines" (CLBVM), which takes load balancing decisions based on global state information. The authors of the research assumed that network load is constant and does not change frequently but it's not applicable in the current cloud environment that changes frequently. Another limitation is the resource type such as memory and I/O that are rarely considered in this work.

Yang et al proposed a Multistrategy prediction mechanism for VM migration to reduce the number of overloaded hosts and avoid unnecessary migration (Yang et al.2011). The authors also adopted a weighted function that considers multiple types of resources. To understand the load of the physical hosts, the authors defined four status domains, light-load domain, optimal domain, warning domain, and overloaded domain. The advantage of this algorithm is, it has the ability to adapt into different situations as it has four different strategies for different host statuses. But there are some drawbacks in this method; this system is based on a recently used set of data but not real time data. This methods uses a protocol called "3 times handshaking protocol" to find the best host for migrating VMs, and also it only considers host PM's CPU utilisation and totally ignored host memory when placing a VM on a host. With this protocol, each host maintains an acceptance queue containing VMs that are waited to be allocated, and this queue updates host utilisation and load increment along with time. These queue updates increase network overhead and it is not suitable for a large scale cloud data centre which hosts thousands of PMs, however this algorithm is suitable for small scale data canter scenarios.

Thiruvenkadam et al. proposed a hybrid genetic based host load aware algorithm for scheduling and optimising VMs. The authors' main objective is minimising the number of VM migrations while balancing the load of physical machines. This method is based on a

metaheuristic approach, which is a high level problem independent algorithmic framework to solve very general classes of problems. The metaheuristic approach to the problem requires more time to process than heuristic algorithms, and also it increases the implementation complexity in a realistic cloud environment.

Thomas Stage et al. proposed a framework called Network-Aware Migration Control and Scheduling Differentiated Virtual Machine Workloads (Stage et al.2009). They used static VM consolidation and this type of method is suitable for small data centres where user demand is predictable but not suitable for large data centres where demand is unpredictable. It also ignored CPU and memory usage statistics of physical hosts when detecting overloaded and underloaded hosts. In addition, the proposed architecture had ignored bandwidth completion between user traffic and VM migration traffic.

Later in this chapter, some heuristic approaches have been discussed including different types of heuristic host detection, and also some VM selection approaches have been analysed and researched. These methods have been proposed to overcome energy consumption and server consolidation issues in data centres, however some of the approaches can be applied when developing algorithms in the proposed framework.

Optimisation of algorithms can be divided into two main categories, these being heuristics and exact algorithms. There are different types of heuristic approach such as metaheuristics and hyper-heuristics. Exact algorithms are designed in such a way that it is guaranteed that they will find the optimal solution in a finite amount of time and this finite amount of time may increase exponentially depending on the dimension of the problem, which increases overhead and complexity. Heuristic approaches are based on previous experience and gathered knowledge about the problems, and many heuristic approaches are very specific and problem dependent unlike exact and metaheuristic approaches.

Therefore, a heuristic approach has been considered in the proposed algorithms in the framework. Heuristic approaches may not guarantee an optimal solution and generally return solutions that are worse than optimal. However, heuristic algorithms usually find “good” solutions in a “reasonable amount of time” A heuristic approach will find good solutions on large size issues, and can be applied to find solutions to VM migration issues in large cloud data centres. They provide an acceptable performance, less overhead and provide solutions at an acceptable cost, because heuristic approaches consume fewer resources compared to other

methods. Heuristic algorithms are more suitable in cloud data centres, because they provide acceptable and fast solutions, less overhead and cost effective solutions.

The proposed VM live migration policy adopted a dynamic load balancing strategy, because static live VM migration depends on the knowledge that has been gathered prior to VM migration. Also it moves VMs among the physical servers without considering the current status of the system. Static load balancing algorithms are not suitable for dynamic cloud environments where the system status changes frequently. Dynamic algorithms respond to the actual current system state in making decisions and move VMs from overloaded servers to an underutilised server dynamically in real time.

From a review of the literature it was found that many researchers considered only one or two resources when implementing load balancing strategies. As an example, Zhao et al's work, they only considered CPU usage and I/O usage in their VM migration policy. Yang et al's work, they only considered host PM's CPU utilisation. Thomas et al's work they considered only network topology and the bandwidth. As an example, if it only considers a single resource usage (CPU utilisation), sometimes a small spike of CPU utilisation will give a false alarm and will trigger unnecessary VM migration causing performance degradation.

It's important to consider a combination of CPU, memory and network bandwidth resources usage when implementing a VM migration policy in order to reduce the number of VM migrations. The reason being a combination of CPU, memory and bandwidth usage data provides a clear state of a PM whether it's overloaded or underloaded plus it prevents unnecessary VM migration. Therefore, the proposed framework multiple resources usage has been adopted for detecting overloaded and underloaded hosts.

Most implementations that have been reviewed in this chapter focused on dynamic load balancing based on various perspectives. But none of the implementations considered network bandwidth when migrating a VM from source to destination. What would happen if the network has insufficient bandwidth to support VM migration load? Is there any method to find out the VM migration load prior to VM migration? How can we provide enough bandwidth for VM migration if there is not enough bandwidth on the network to support VM migration traffic?

The next chapter will present the detailed design of the proposed Framework and it will answer all the above questions.

Chapter 5: Proposed Framework

5.1 Introduction

In the literature review it proved that there is a correlation between VM load and the available bandwidth, both parameters contribute for network contention in live VM migration. When more VMs are migrating, it increases the network overhead and consumes more bandwidth, which increases the total migration time. On the other hand, when the data centre network is overwhelmed with other network traffic such as user traffic, it creates a competition for the bandwidth between VM migration traffic and user traffic. When there is not enough bandwidth to support VM migration, it increases the total VM migration time and may create network bottlenecks.

To address that problem, the proposed framework provides an essential ground for understanding the total network contention due to VM migration traffic and other network traffic, mainly user traffic. According to the insight gained from the literature review, a comprehensive intelligent, dynamic, and automated framework for live VM migration was proposed.

The proposed framework is based on the approach to the problem that is presented in the problem statement. The main problem has been divided into seven sub problems which are shown below.

1. Finding Overloaded hosts? It's important to determine overloaded hosts in a data centre in order to prevent it from failing and creating SLA violation.

2. Finding Underloaded hosts? Determining underutilised hosts in a data centre would help to find possible candidates for migrating VMs.

3. Finding the best VMs to migrate? Once decided which PM is needed to perform VM migration, then identification is required as to which virtual machine or machines to move. The problem lies in determining the best subset of VMs to migrate that will provide the best beneficial system reconfiguration.

4. Finding the best host for VM migration? Determining the best host for migrating VMs will be an essential aspect of the framework.

5. When to migrate a VM? Once the overloaded hosts are found, it's important to move VMs from overloaded hosts to underloaded hosts to avoid performance degradation. A crucial decision must be made to determine the best time to trigger VM migration.

6. Finding the available bandwidth on the network as well as required minimum bandwidth for migrating VMs? The Bandwidth Predictor in the Central Controller component of the framework is responsible for finding the available bandwidth on the network and calculating minimum required bandwidth for VM migration.

7. Finding the amount of user traffic on the network? The User Traffic Controller component of the Central Component is responsible for finding the user traffic on each VM.

8. Finding the best VM to control the user traffic in order to provide required bandwidth for migrating VMs? If there is not enough bandwidth on the network to support VM migration, the next step is finding the best VMs to control user traffic in order to provide adequate bandwidth for VM migration.

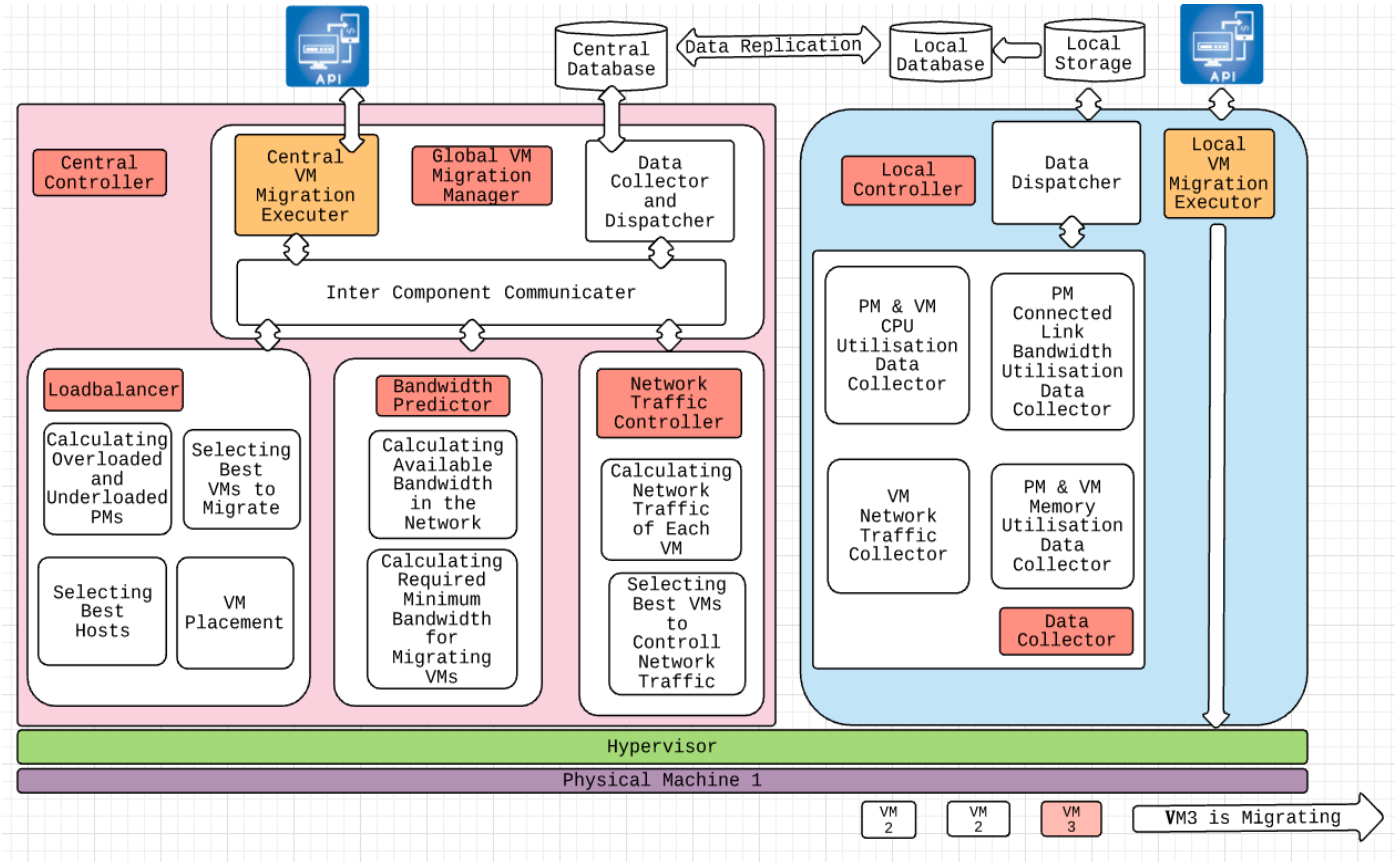


Figure 5 .1: Proposed Bandwidth Aware Dynamic VM Migration Framework

5.2 Framework Detailed Design

The aim of this project is to provide a network bandwidth aware intelligent dynamic automated framework for live VM migration. The proposed framework provides an infrastructure required for monitoring VMs and PMs, collecting resources from usage and network traffic data, transmitting messages and commands between the framework components, and invoking VM live migration. The framework's infrastructure is agnostic to correspondent customized algorithms and equations, which are related to each component. The proposed algorithms are later presented in the Implementation chapter.

The proposed framework (Figure 5.1) is equipped with two main components, Central Controller and Local Controller. The Central Controller and Local Controller communicate, accept requests and issue commands through an application programming interface (API). The Data collector component in the Local Controller periodically collects resource usage of VMs and PMs in the cluster, and then the collected data will be stored in the local file-based data store and local database, then later periodically transferred it to the central database. The

Central Controller fetches data from the central database and is responsible for making decisions by invoking proposed algorithms.

5.2.1 Central Controller

The proposed framework has been adopted a centralised load balancing approach, The Central Controller is deployed on the controller host node along with the Local Controller. There is only one central controller for a cluster, which makes global management decisions and initiating VM live migrations. However, we can deploy the second Central Controller on a backup server in the same cluster to improve reliability. The following are the main components of the Central Controller, and each component of the framework is described in the next section. The Central component is shown in the figure 5.2.

- Loadbalancer
- Bandwidth Predictor
- Network Traffic Controller
- Global VM Migration Manager

5.2.1.1 Load Balancer Component

The load balancer component is responsible for balancing the load on physical machines in the data centre, it resides in the Central Controller and it's responsible for making following decisions;

- Detection of overloaded PMs
- Detection of underloaded PMs
- Selection of best VMs to migrate
- Selection of best hosts for migrating VMs
- VM Placement

The Load balancing component communicates with the Bandwidth Predictor and Network Traffic Controller through the Inter Component Communicator component.

Detection of Overloaded PMs

Each PM periodically executes the new overloaded detection algorithm to identify overloaded PMs in the data centre, the algorithm will be based on setting a static CPU,

memory and bandwidth utilization threshold to detect overloaded PMs. When the algorithm is invoked, it compares the current CPU, memory, and bandwidth utilization of the host PM with the defined threshold. The algorithm finds overloaded hosts under the following conditions.

- When all three resources' utilisation of a host (CPU, memory and bandwidth utilisation) exceed over the threshold.

Detection of Underloaded PMs

The algorithm will be based on setting a static CPU, memory, and bandwidth utilization threshold to detect underloaded PMs, When the algorithm is invoked, it compares the current CPU, memory, and bandwidth utilisation of the host PM with the defined threshold and it detects an underloaded host under the following conditions.

- When all three resources' utilisation of a host (CPU, memory and bandwidth utilisation) are less than or equal to the threshold.

Selection of Best VMs to Migrate

Once an overloaded host is detected, it's important to determine the best set of VMs or a VM for migration in order avoid SLA violations. This problem is resolved by applying the VM selection algorithm. The algorithm first selects the VMs that have the least memory utilisation in order to reduce VM migration time. Then out of selected VMs, it selects the VMs that have maximum bandwidth in order to reduce the bandwidth competition between user traffic and VM migration traffic, because it assumes that bandwidth utilisation is low due to less traffic. Then out of selected subset of VM, it selects the VMs that have the maximum utilisation in order to maximally reduce overall CPU utilisation of the host.

Selection of Best Host for Migrating VMs

This problem involves selecting the best host machines for migrating VMs and is resolved by placing the VMs on underloaded machines.

VM Placement

In general, VM placement could be seen as a bin packing problem with variable bin sizes and prices. This problem can be resolved by applying the proposed VM placement algorithm which is presented in the Implementation chapter.

5.2.1.2 Bandwidth Predictor

Bandwidth Predictor is responsible for calculating available total bandwidth in the network and minimum required bandwidth that is needed for migrating VMs. Every physical connected server in the cluster will collect resources utilisation data of each PM and VM and send it to the central database. Then it calculates the available bandwidth and the required minimum bandwidth for migrating VMs. The framework has a clear picture of the available bandwidth on the entire network. The main function of the Bandwidth Predictor is, it calculates required minimum bandwidth for migrating VMs.

5.2.1.3 Network Traffic Controller

This component is responsible for controlling user traffic on the network in order to facilitate the required minimum bandwidth for VM migration. This component does the following calculations,

- Calculating user traffic on each VM
- Calculating the minimum amount of user traffic to be controlled
- Select the best VMs to control user traffic.

5.2.1.4 Global VM Migration Manager

The Global VM Migration Manager consists of three components, which are Data Collector and Dispatcher, Central VM Migration Executer and Inter Component Communicator. Each component is described below.

Data Collector & Dispatcher: This component is responsible for fetching data from the central database and dispatching it to the relevant components in the framework, and also it is responsible for sending data from components to the central database.

Central VM Migration Executer: It accepts VM migration execution commands from the Load balancer, Bandwidth Predictor and Network Traffic Controller component and executes those commands through an application programming interface (API).

Inter Component Communicator: This component acts like a mediator among Central Controller components (Loadbalancer, bandwidth predictor, network traffic controller, central VM migration executor, and data dispatcher) by facilitating communication between different components in the Central Controller. The flow chart of the Central Controller is shown in the figure 5.3.

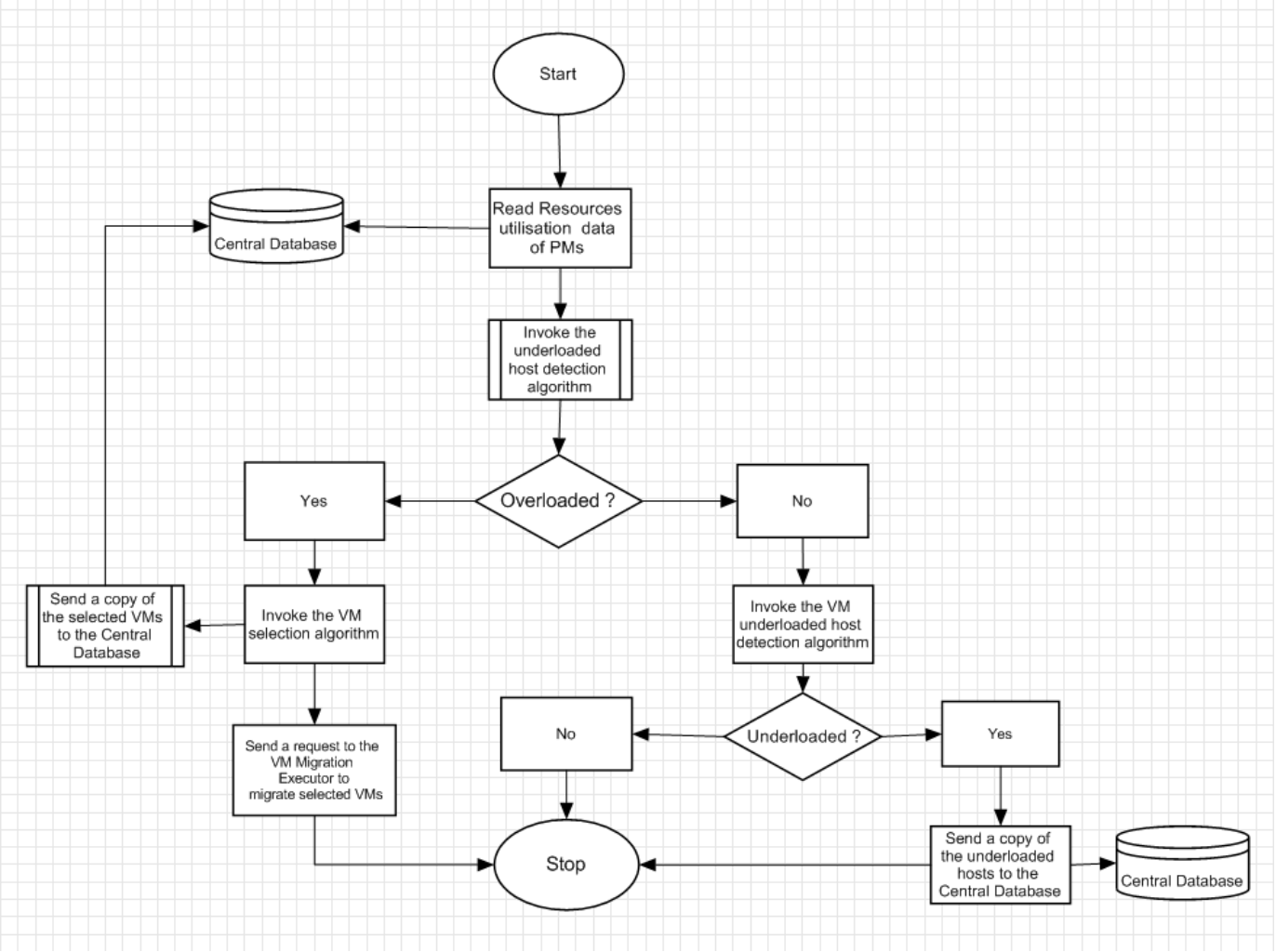


Figure 5.3: Central Controller Component Flow Chart

5.2.2 Local Controller Components

A Local Controller is deployed on every PM in the cluster, but there is one Central Controller for a cluster. The Local Controller (Figure 5.4) is equipped with three main components, which are:

1. Data Collector
2. Local VM Migration Executor
3. Data Collector & Dispatcher

Data Collector: This component periodically collects CPU and memory utilisation, bandwidth on the connected link, and user traffic on each PM, and it also collects the CPU utilisation by the hypervisor. Collected data will be saved in the local file based storage.

Data Dispatcher: The dispatcher acts as the central hub that manages all data flow in the Local Controller. It has a simple mechanism for transferring data from the local file based data storage and to the central database.

Local VM Migration Executor: This component is responsible for passing VM migration execution commands to the local Hypervisor. It receives VM migration execution commands from the Central VM Migration Executor through an application programming interface (API).

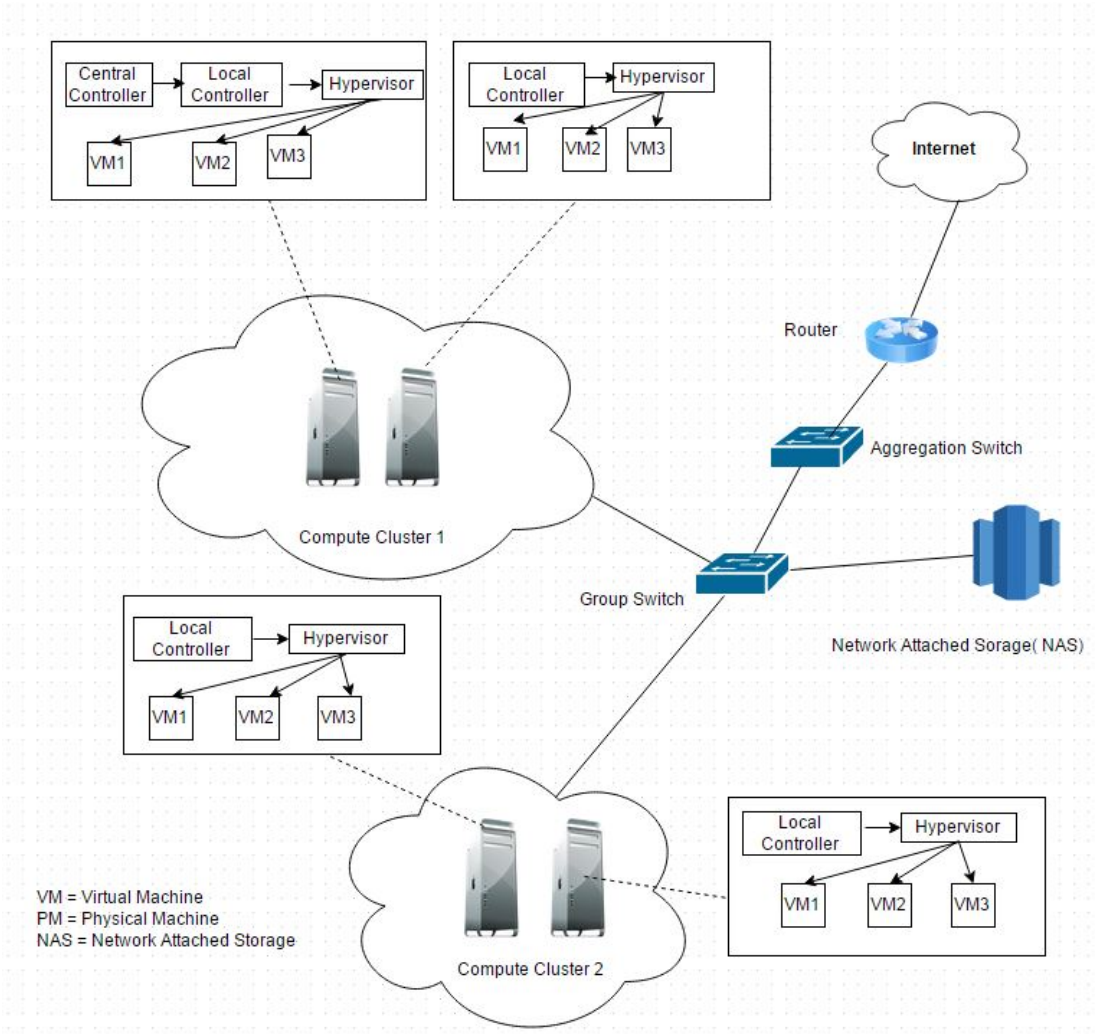


Figure 5.5: Deployment of Central and Local Controllers in a Cluster

The implementation of the proposed framework in a real data centre scenario is shown in figure 5.5. In a typical data centre, a cluster may have hundreds to thousands of physical servers.

5.2.3 Data Storages and Databases

There are three main data storage systems, central database, the local file-based data storage and the local database.

Local File-Based Data Storage: It's deployed on every PM and it can be used for temporarily cashing the resources' usage and other statistics that are collected from the host. The data collector temporarily stores unstructured resource usage data locally on Local File-

Based Data Storage, and later it is organizes it as structured data and transfers it to the Local Database.

Databases

There are two types of databases in the system: Local and Central. Both the Local and Central databases use the same database schema and both are used to store historical data on resources usage as well as hardware characteristics of PMs and VMs. The Local database temporarily stores resources usage data of PMs and VMs, and then periodically transfers the data to the Central database. The purpose of having the Local database is to prevent flooding of the data centre network by database traffic

The Central database is used for storing historical data centrally and the Central Controller of the framework will use the latest data (according to the timestamp) in the database to make decisions.

Local Database (LD): The Local database is connected to every PM for storing structured data that are collected from the host, and then at regular intervals it sends a copy to the Central Database through data replication. This reduces the number of queries being sent to the Central Database over the network.

Central Database (CD): The central database stores historical data of every PM, and it replicates data with local databases which are populated by the data collectors of each PM.

These are use cases when data is retrieved from the Central database, it is first used by Data Collector and Dispatcher of the Central Component to fetch resources' usage data from the CD, because it needs to calculate overloaded and underloaded hosts. The second use case is when the Central Controller needs to select VMs that need to be migrated. The third is VM Placement because the VM placement algorithm requires information on the resources' consumption of all hosts including user traffic in order to make VM allocation decisions. The data storages and databases of the proposed framework are shown in the figure 5.6.

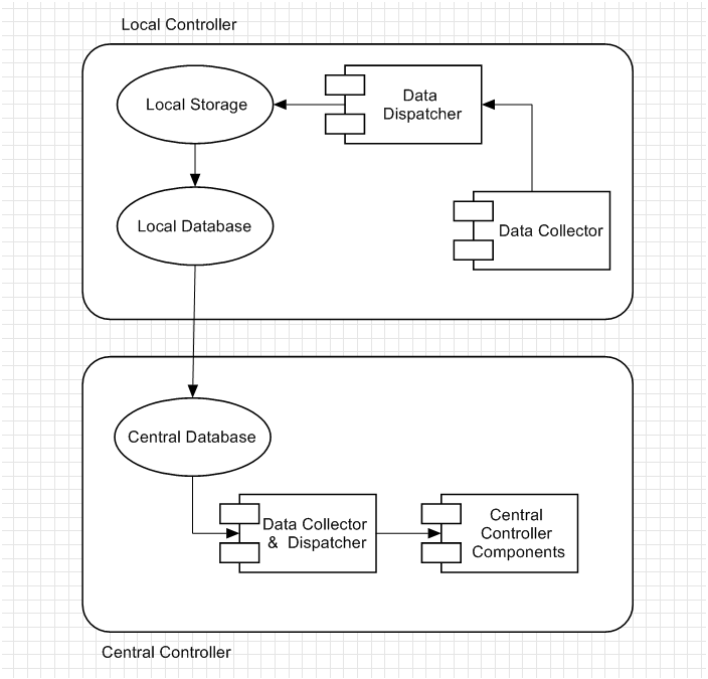


Figure 5.6: Data Storages and Databases

The second use case of the Central Database is when it is queried periodically to check the status of the system. When the overload detection algorithm recognises an overloaded host in the cluster, it computes a new placement of VMs on the host. When this occurs, four algorithms in the Central Controller will require information on the resources consumption of all hosts in order to make global decisions. Therefore, the Central Controller queries the CD every time as it needs maximum resources usage information of hosts and VMs in order to make decisions. The database schema of the CD contains four main tables as shown in the table 5.1, 5.2, 5.3 and 5.4.

Field	Type
id	Integer
hostname	String (255)
cpu_mhz	Integer
cpu_cores	Integer
cpu_totalcapacity	Integer
allocated_ram	Integer

allocated_bandwidth	Integer

Table 5.1: The Hosts Table (Physical Machines)

Field	Type
vm_id	Integer
uuid	Integer

Table 5.2: The VMs Table

Field	Type
id	Integer
host_id	Integer
timestamp	DateTime
cpu_mhz	Integer
memory_usage	Integer
bandwidth_usage	Integer

Table 5.3: The Host Resources Usage Table

Field	Type
id	Integer
vm_id	Integer
timestamp	DateTime
cpu_mhz	Integer
memory_usage	Integer

bandwidth_usage	Integer

Table 5.4: VM Resource Usage Table

The hosts table stores information about physical machines in the cluster, such as the host names, CPU frequency of a physical core in MHZ, the number of CPU cores and the total capacity of the host, amount of RAM and bandwidth. The VMs table stores the UUIDs (universally unique identifier) of VMs assigned by the cloud platform. The host resource usage and VM resource usage tables store resource consumption data over time by hosts and VMs respectively.

5.3 Chapter Summary

This chapter has presented the proposed Network Bandwidth-aware Intelligent and Dynamic framework for VM migration in Cloud .And also it has given a detailed design of the framework and its functionality of each component. The framework is made up of different components, and those various components provide different functionalities and services to the framework. The two main components of the framework are the Central Controller and Local Controller, and they communicate with each other through an Application Programming Interface (API). The defining characteristics of the framework are its ability to learn different types of resources' utilisation of PMs and VMs, calculating the network's available bandwidth, calculating minimum required bandwidth for migrating VMs and facilitating required minimum bandwidth for VM migration by controlling user traffic. It also has the ability to dynamically migrate the VM from overloaded hosts to the underloaded hosts to balance the load of PMs in a data centre. The next chapter presents the Implementation of the proposed framework.

Chapter: 6 Implementation and Evaluation

This chapter starts with an introduction and then it moves to the implementation of corresponding algorithms where it discusses detection of overloaded hosts, underloaded hosts, VM selection and VM placement algorithms. A collection of proposed four algorithms are called a VM migration policy, which was implemented using the CloudSim. Two benchmark policies in CloudSim, ThrRs and ThrMMT have used in the simulation to evaluate the proposed VM migration policy. The results of the experiment have evaluated using five performance metrics, which were discussed in this chapter. This chapter (Chapter 6) presents the detailed implementation of the proposed VM migration policy, evaluation and comparison of results.

6.1 Introduction

The aim of this research is to build a bandwidth-aware, intelligent, dynamic, and automated framework for live VM migration; it guarantees a required bandwidth for VM migration traffic by controlling user traffic on the network. The framework contains several components and each component is responsible for carrying out a specific task, the Loadbalancer component consists of four algorithms which are the core algorithms in the framework. The detailed proposed framework was presented in chapter 5.

The collection of the four proposed algorithms is called a VM Migration Policy, and the proposed VM migration policy has four steps which are shown in the figure 6.1.

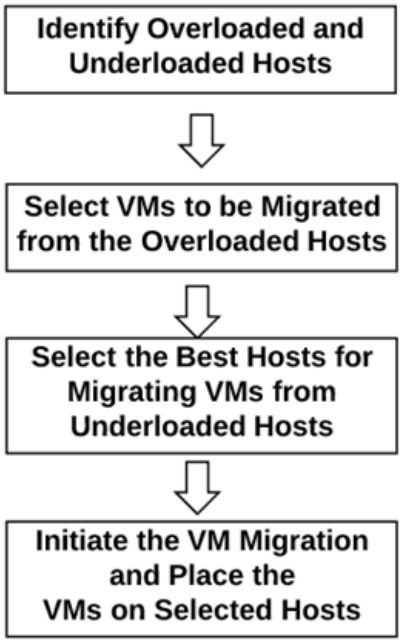


Figure 6.1: The Proposed VM Migration Policy

Ideally the algorithms would have been evaluated on a real cloud infrastructure. However, it is not feasible to conduct a large-scale experiment on a real cloud infrastructure due to the complexity, cost, and lack of benchmark software, workload traces, performance metrics, and evaluation methodology. Therefore, most researchers develop their own solutions for evaluating new algorithms.

The functions of Bandwidth Predictor, Network Traffic Controller and Global VM Migration components will be implemented in the VM placement algorithm. The figure 6.2 shows the VM migration policy, corresponding algorithms and components of the framework.

In this research, CloudSim will be used to implement and evaluate the four proposed algorithms in the VM migration policy. Collecting statistical data of VMs and PMs can be implemented in the CloudSim (CloudSim Classes), in addition some methods of CloudSim can mimic the functions of the Local Controller component of the framework (Calheiros et al.2011). The functions of the Load Balancer component will be implemented in the following algorithms.

- Host overloaded detection algorithm

- Host underloaded detection algorithm
- VM Selection algorithm
- VM Placement algorithm

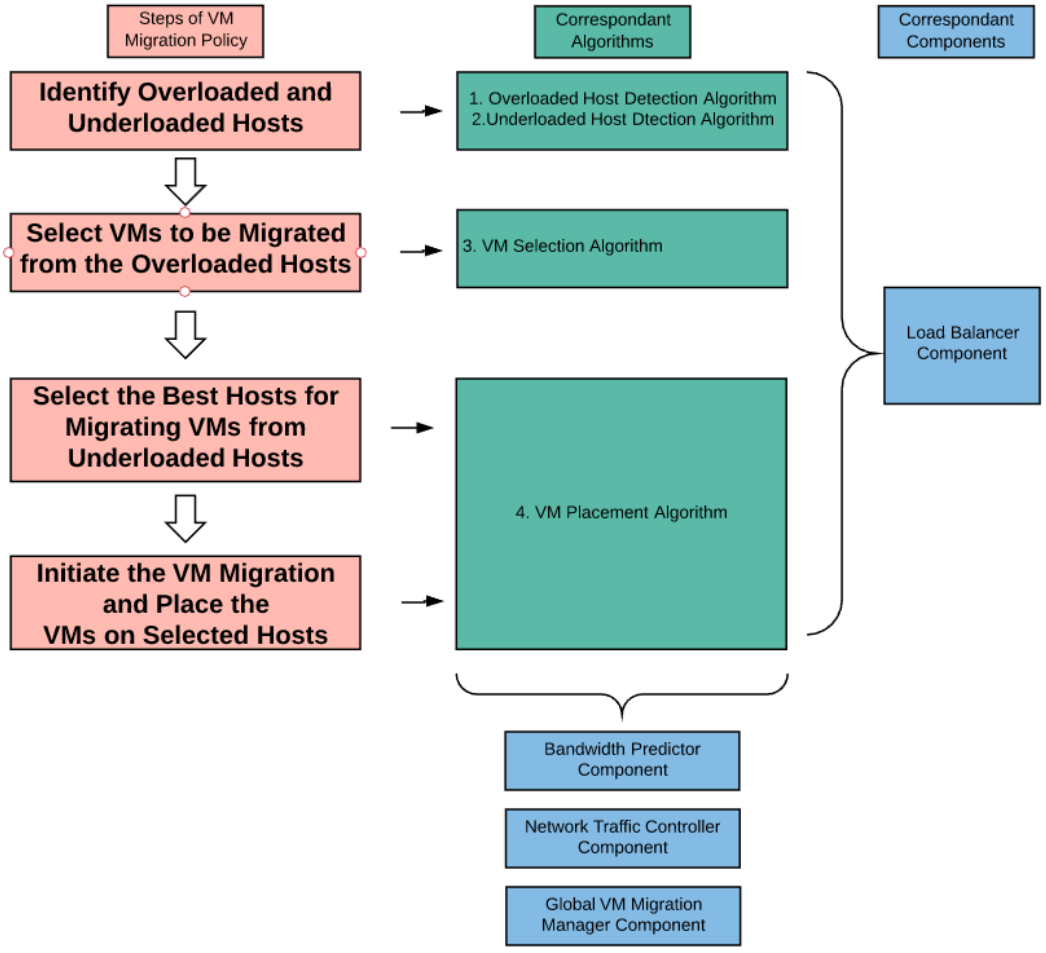


Figure 6.2: The VM migration policy, correspondent algorithms and components of the framework

6.2 Implementation of Correspondent Algorithms of the Framework

In this research a Heuristic based approach is adopted to respond to an overloaded or underloaded host, VM selection, and VM placement algorithms. As it stated in the Chapter 5, heuristic is set of constraints that aim at finding a good solution for a particular problem in a reasonable amount of time. The solution might not the best one, but the approach finds a reasonable solution in a very short period of time. In addition, heuristic algorithms are easier to implement in comparison to meta-heuristic algorithms. Heuristic algorithms run fast and

they are suitable for online scheduling that requires the system to respond in reasonable amount of time.

6.2.1 Detection of Overloaded and Underloaded PMs

Continuously monitoring resources utilisation on PMs, overloaded and underloaded machines can be detected. When the resources capacity of the host is completely or almost completely utilised, VMs running on the overloaded host will experience a shortage of resources and this will cause performance degradation. Degrading performance will directly influence the QoS (Quality of Service), and also if the load changes dramatically and there are not enough resources, there will be frequent SLA (Service Level Agreement) violation problems.

It's desirable that overloaded hosts are detected dynamically rather than statically, because statically forecasting the load of a PM is very difficult in a cloud computing environment. As most cloud-based applications have a fluctuant load which leads to complex behaviours in resources usages because their intensity and composition change over time. Interference among VMs hosted on the same PM leads to complex resources usage behaviors as they compete for various types of resources. As an example, figure 6.3 shows a real world scenario where Twitter experienced dramatic load fluctuations on US president Barack Obama's inauguration day, 2009 (Hu et al. 2014).

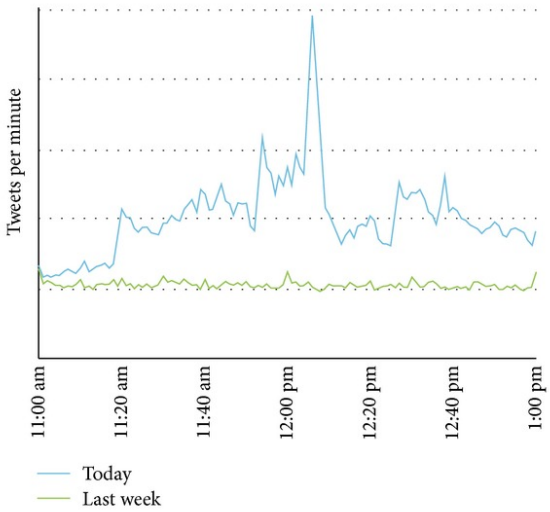


Figure 6.3: Load of Twitter on Obama's Inauguration (Hu et al. 2014)

In the proposed overloaded and underloaded hosts' detection algorithms, the system observes a multiple resources utilisation of PM for overload detection, because it reduces unnecessary VM migration. As an example, if it only considers a single resource usage (CPU utilisation),

sometimes a small spike of CPU utilisation will give a false alarm and will trigger unnecessary VM migration causing performance degradation. In order to ensure that a small transient spike does not trigger unnecessary migration, multiple resources usage has to be considered in this algorithm.

Therefore, the proposed overloaded and underloaded host detection algorithms monitor CPU, memory and bandwidth utilisation of hosts in real time. The local controller reads the behaviour of VMs and PMs, and it collects PM and VM bandwidth utilisation data, user traffic, PM memory utilisation and PM CPU utilisation every 10 minutes and records it in the local file based data storage. Then the local data storage and the central database synchronises data at regular intervals, the duration of the interval can be set up by the administrator according to the data centre's requirements.

6.2.1.1 Overloaded Host Detection

Complex underloaded detection strategies can be applied here, but for the purpose of simulation, a simple threshold-based strategy has been used. First it executes the algorithm on PMs and it compares the current CPU utilisation of the host with the defined threshold (80%) If the CPU utilisation of the host exceeds the threshold, then the algorithm compares the current available bandwidth with the defined threshold (80%), if any host exceeds CPU and the available bandwidth threshold, then it compares the current memory utilisation of the PM with the defined threshold (80%) If the host's all three resources utilisation values exceeded the thresholds, then it will detect an overloaded host. The overloaded host's detection algorithm takes the following steps.

1. Finding the host CPU utilisation
2. Finding the host bandwidth utilisation from the available bandwidth
3. Finding the memory utilisation of the host

Step 1. Calculating Host CPU Utilisation Percentage from the Total CPU Capacity

A processor or CPU (central processing unit) is the primary component of a computer, which is responsible for executing instructions of a computer programme. A Core is an operational unit inside a CPU similar to a processor, generally a core can handle one thread at a time but with the help of technologies like Hyper-Threading, a core can handle two threads concurrently. The number of threads in a core is equal to the number of logical processors.

The latest physical servers are equipped with multi-core CPUs. “A multicore processor is typically a single processor which contains several cores on a chip” (Wang et al, 2010). A multi-core CPU has *n* number of cores and each core having *m* MIPS (million instructions per second) of multiple cores is modelled as a single core with the total capacity *nm* MIPS. The performance of a CPU can be measured in millions instructions per seconds (MIPS). As an example, VMware ESXI vSphere 6.5 host offers a maximum of 1024 VMs per host, 578 logical CPUs per host, and 4096 virtual CPUs per host according to VMware (VMware, 2017)

A data centre may be equipped with heterogeneous PMs that have different processing power. We consider multiple CPU cores in a physical machine, and we define *n* number CPU cores $(\text{cpu core } 1, \text{cpu core } 2 \dots \dots \dots \text{core n})$ on a physical machine (PM) in a cluster. The utilisation of j^{th} core of the physical machine is CPU_j . Total CPU utilisation percentage of j^{th} core of the physical machine in the cluster can be represented by $PMcpu\ Utilisation$, and the final value must be multiplied by 100 to get a percentage; the equation is shown in *Eq 6.1*.

$$PMcpuUtilisation = \frac{\sum_{j=1}^{n} CPU_j\ (MIPS)}{Total\ CPU\ \ capacity\ of\ the\ host\ (MIPS)} \times 100 \qquad Eq(\ 6.1)$$

Equation 6.1 shows how to calculate the CPU utilisation percentage from the total capacity of a multicore physical machine.

As mentioned in the literature review, considering only a single resource threshold to detect overloaded hosts would give false alarms and it might trigger unnecessary VM migrations. Therefore, this research considered multiple resources utilisation (CPU, memory and bandwidth), while setting up upper and lower utilisation thresholds for overloaded and underloaded host detection.

The study of Srikantaiah et al. has shown that the performance of a physical host can be degraded when the CPU utilisation is higher than 70 % (Srikantaiah, 2009). However, one of the most important aspects of the proposed framework is reducing number of VM migrations in a data center, because unnecessary VM migrations drain data centre resources which also degrade performance. Therefore, CPU, memory and bandwidth utilisation threshold values are set at 80 % in order to reduce the number of overloaded hosts, which will then reduce the number of VM migrations in a cluster. Hosts which exceeded the threshold are regarded as overloaded and these hosts need to perform VM migration to prevent potential SLA violation.

Step2. Calculating the Host Bandwidth Utilisation Percentage from Available Bandwidth

The next step is calculating the host bandwidth utilisation % from the available bandwidth, but before that it is necessary to calculate the available bandwidth on the network.

Calculating Available Bandwidth on the Network

The Local Controller of the proposed framework collects bandwidth utilisation data of all PMs and VMs in the cluster, and it records them in the local storage. Then the data will be sent to the central database at regular intervals. Calculating available bandwidth on the network is a function of the Bandwidth Predictor component of the proposed framework, then the available bandwidth on the network is passed to the algorithms which reside in the Loadbalancer component.

A data centre may be equipped with hundreds of PMs which are hosting thousands of VMs. We define *m* number of PMs in the data centre; the bandwidth utilisation of each PM is A_k: the total bandwidth utilisation (TotalPMBdwutlisation) of all PMs in the cluster can be calculated using the equation below (Eq 6.2).

$$TotalPMBdwutlisation = \sum_{k=1}^{m} A_k \qquad Eq\ (6.2)$$

Now we have the total bandwidth utilisation of all PMs in the cluster, the available bandwidth in the network can be calculated using the formula below (Eq 6.3).

$$Available\ Bandwidth = Total\ Network\ Bandwidth - TotalPMBdwutlisation \qquad Eq\ (6.3)$$

As we have the available bandwidth on the network, the PM's bandwidth utilisation percentage from the available bandwidth of the network can be calculated using the following formula which is shown below.

$$Host\ bandwidth\ utilisation\ \% = \frac{PM\ \ Bandwidth\ \ \ Utilisation}{Aailable\ bandwidth\ on\ the\ netork} \times 100 \qquad Eq\ (6.4)$$

Step 3. Finding the Memory Utilisation of the Host

As this experiment uses the CloudSim for simulation, RAM utilisation of each host will be obtained by using a CloudSim method (Calheiros et al.2011).

<u>Proposed Overloaded Host Detection Algorithm</u>

Input: *hostList, each host CPU Core utilisation data, Host bandwidth utilisation data,* $Total\ CPU\ capacity\ of\ each\ Pm$ *, host memory utilisation data, Total Network Bandwidth Capacity*

Output: *Whether the host is overloaded*

1. *n=number of cores in a server*
2. *Get the used bandwidth of each host from the database*
3. *Calculate Total Bandwidth Utilisation using (Eq 6.2)*
4. *Calculate the available bandwidth on the network (Eq 6.3)*
5. *Get the memory utilisation (memorytUtil) of each host from the database*
6. *Get the hostCPU utilisation data(*CPU_j*) from the central database*
7. *CPU* $threshold$ *= 80 %*
8. $memoryUtilThreshold$ *= 80 %*
9. *BandwidthThreshold. = 80 %*
10. ***for*** *host in hostList* ***do***
11. *Calculate the CPU utilisation percentage (PMcpuUtlisation)from the total CPU capacity of each host*
12. $PMcpuUtlisation = \frac{\sum_{j=1}^{n} CPU_j \text{ (MIPS)}}{Total\ CPU\ capacity\ of\ the\ host(MIPS)} \times 100$
13. *Calculate the host bandwidth utilisation from available bandwidth*
14. $Host\ bandwidth\ utilisation\ \% = \frac{PM\ Used\ Bandwidth}{Available\ bandwidth\ on\ the\ network} \times 100$
15. ***If*** *the host CPU utilization(PMcpuUtlisation)is* $\geq CPUthreshold$
16. $and\ host\ bandwidth\ utilisation\ (HBdwUsage) \geq BandwidthThreshold$ *, and memorytUtil* $\geq memoryUtilThreshold$
17. ***then*** *Mark the host as overloaded*
18. ***else***
19. ***continue***
20. ***return*** *set of overloaded hosts*
21. *Send the overloaded hosts list to the central database*
22. ***end***

6.2.1.2 Host Underloaded Detection

The purpose of the algorithm is finding the underloaded PMs by monitoring the CPU, memory and bandwidth utilisation of the PM and then comparing it with a threshold. The algorithm uses the same three steps like in the overloaded host detection algorithm but only the threshold values have changed in the underloaded host detection algorithm. Just as in the overloaded detection algorithm, multiple resources utilisation has been considered here. By doing this, it provides a clear picture of the server's resources utilisation and false alarms are prevented. The lower threshold values for CPU and memory utilisation are set at 30% and the threshold for bandwidth utilisation is set at 50 %.

The idea of setting up a lower threshold value of 30 % for CPU and memory utilisation enables identification of underloaded or less utilised physical machines in a cluster. Since CPU and memory are the most important resources in a server, the identification of low resources (CPU and memory) utilised servers in a cluster means these servers can be shut down or act as hosts for migrating VMs from overloaded hosts. This can help the data centre to utilise resources in an efficient manner saving money and energy.

When multiple applications are running on VMs, the communication needs to be synchronised with each virtual machine at regular intervals to maintain the consistent state of the VM (Kejiang Ye et al. 2012). If the bandwidth threshold is set at 30 %, then it will detect all servers whose bandwidth utilisation below the threshold but sometimes the bandwidth utilisation of servers may exceed the threshold due to intensive communication between VMs even their CPU and memory utilisation fall below the threshold. Therefore, the lower threshold for bandwidth utilisation is set at 50 % to avoid false alarms.

The proposed algorithm first executes the algorithm on PMs and it compares the current CPU utilisation of the host with the defined threshold (30 %). If the CPU utilisation of the host below the threshold, then the algorithms compare the memory with the defined threshold (30%), if both resources are below the threshold then it compares bandwidth usage percentage with the defined threshold (50%), if all three resources are below the threshold, it will mark the PM as an underloaded host.

Proposed Underloaded Host Detection Algorithm

Input: *hostList, each host CPU Core utilisation data, Host bandwidth utilisation data , $Total\ CPU\ capacity\ of\ each\ Pm$, host memory utilisation, Total Network Bandwidth Capacity*

Output: *Whether the host is underloaded*

23. n=number of cores in a server
24. Get the used bandwidth of each host from the database
25. Calculate Total Bandwidth Utilisation using (Eq 6.2)
26. Calculate the available bandwidth on the network (Eq 6.3)
27. Get the memory utilisation (memorytUtil) of each host from the database
28. Get the hostCPU utilisation data(CPU_j) from the central database
29. CPU $threshold$ = 30 % (from the total capacity of the PM)
30. $memoryUtilThreshold$ = 30 % (of PM memory)
31. BandwidthThreshold.= 50 %
32. ***for*** *host in hostList* ***do***
33. Calculate the CPU utilisation percentage (PMcpuUtlisation)from the total CPU capacity of each host
34. $PMcpuUtlisation\ \% = \frac{\sum_{j=1}^{n} CPUi_j \ \text{(MIPS)}}{Total\ CPU\ capacity\ of\ the\ host\ (MIPS)} \times 100$
35. Calculate the host bandwidth usage from available bandwidth
36. Host bandwidth utilisation % = $\frac{PM\ Used\ \ Bandwidth}{Available\ bandwidh\ on\ the\ network} \times 100$
37. ***If*** *the host CPU utilisation on($PMcpuUtlisation\ \%$) is $\geq CPUthreshold$*
38. and host bandwidth usage ($HBdwUsage$) $\geq$ BandwidtThreshold , and
39. memorytUtil $\geq$ $memoryUtilThreshold$
40. ***then*** *Mark the host as overloaded*
41. ***else***
42. ***continue***
43. ***return*** *set of overloaded hosts*
44. ***End***

6.2.2 VM Selection

The proposed framework determines overloaded and underloaded hosts, then it needs to select which VMs to migrate and where. Once an overloaded PM has been detected, it is important to select what VMs are the best to be migrated from the host. This issue can be resolved by VM selection algorithms, different types of VM selection algorithms have been analysed and discussed thoroughly in the chapter 4. As an example, the simplest of VM selection algorithm is Random Selection Algorithm, which selects VMs randomly from set of VMs that are allocated to the host. Many different types of VM selection algorithms have been discussed and analysed in chapter 4.

Proposed Minimum VM Memory, Maximum VM CPU and Maximum Available Bandwidth VM Selection Algorithm

Reducing the migration overhead is important in VM live migration (amount of data transferred), The VM memory migration technique depends on the hypervisor technology that is being used. The proposed algorithm first selects the VMs that have the least memory utilisation in order to reduce the total VM migration time. As an example a VM that has 16GB of memory might take sixteen times longer migration time than a VM with 1GB memory.

Then it selects the VMs that have the maximum available bandwidth, because those VMs get less user traffic and bandwidth competition between user traffic and VM migration traffic is reduced. Then out of the selected subset of VMs, the algorithm selects the VM that has maximum CPU utilisation to maximally reduce overall CPU utilisation of the host.

Input: *VMList, vmsCPUutilisation,vmsRamUtilisation ,vmAvailableBandwidth,*

Output: *SelectedVm List*

1. *Get the vmRamUtilisation from the database*
2. *Get the vmCPUutilisation from the database*
3. *Get the vmAvailableBandwidth from the database*
4. *minRam ⟵ min (vmRamUtilisation)*
5. *maxCpu ⟵ max (values from vmCPUtilisation)*
6. *maxBdw ⟵ max (values from vmAvailableBandwidth)*
7. *SelectedVm ⟵* ***None***
8. ***for*** *vm, in the VMList* **do**
9. ***if*** *vmRamUtilisation > minRam* ***then***
10. *select it as a candidate for VM migration*
11. ***Continue***
12. ***end for*** *Out of selected subset of VMs that has minRam* ***do***
13. ***If*** *vmsCPUutilisation > maxCpu* ***then***
14. *select it as a candidate for VM migration*
15. ***Continue***
16. ***end for*** *Out of selected subset of VMs that has minRam and maxCpu* ***do***
17. ***If*** *the vmAvailableBandwidth > maxBdw* ***then***
18. *selectedVM for migration ⟵* ***True***
19. *Remove the VM from the VMList*
20. *Send the migration list to the database*
21. ***end if***
22. ***end for***
23. ***return*** *SelectedVm List*

6.2.3 VM Placement

The VM Placement issue in the proposed framework can be seen as a bin packing problem with different bins sizes where bins represent physical hosts, and the size of the bin represents the available CPU capacity of the host. The items that need to be allocated to the bins represent different VMs. Different types of VM placement algorithms have been discussed in the Chapter 4. As we discussed in the literature review, VM live migration increases network overhead and VM live migration will compete to the bandwidth with user traffic. When more VMs are migrating, the network overhead is increased and more bandwidth consumed, which increases the total migration time.

VM migration requires a substantial amount of bandwidth between the source and the destination host. VM migration time depends on the VM load and the bandwidth provided. The assigned network bandwidth rate for any VM migration is a key factor in determining the migration latency and downtime. VM downtime is unavoidable, and it depends on the VM memory migration technique. However, by employing different types of optimising methods the VM migration downtime can be significantly reduced.

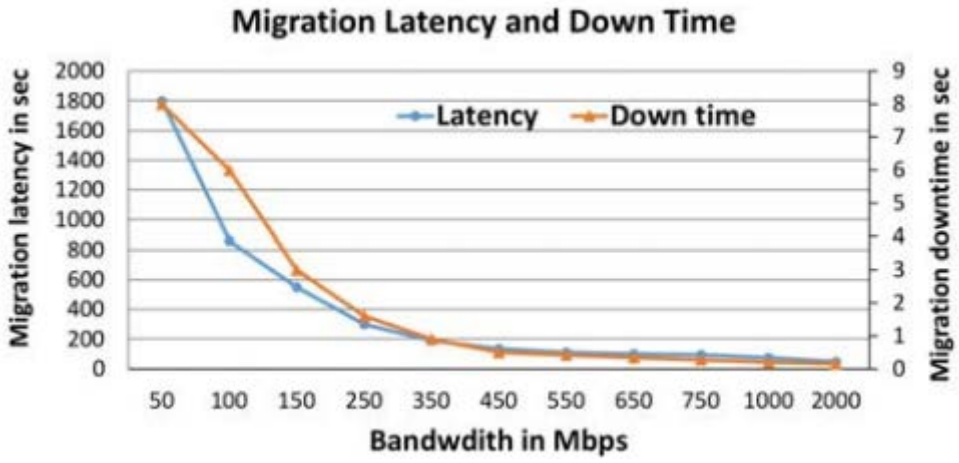


Figure 6.4: VM Migration Characteristics of a Virtual Server that Running a TPC-W Benchmark application (Mandal et al.2013)

Figure 6.4 shows that typical migration characteristic of a VM (VM server) with 1.5 GB memory and 6GB hard disk storage running a TPC-W benchmark application (which is a web server and database performance benchmark that proposed by Transaction Processing Performance Council). The server was assigned bandwidth from 50 Mbps – 2Gbps, and the

bandwidth can be provisioned higher than 2Gbps. The primary y-axis of the graph shows migration latency, and the secondary y-axis shows migration downtime in seconds. With increasing bandwidth, the VM migration time varies from 1800 seconds to 50 seconds, and the downtime varies between 8 to 0.15 seconds (Mandal et al.2013). It shows that VM migration needs to be completed within a specified duration of time ensuring that the VM migration time and downtime is less than or equal to the specified value.

The term latency refers to any of several types of network delays typically incurred in the processing of network data. Network delay is the time taken for a bit to travel from one end to another which could be measured in seconds or fraction of seconds. There are various types of delays might occur on a network due to various factors such as network hardware limitations, overhead in the communication link, a slow transmission rate in the link, and the position or the distance from the hosts from each other. Different types of delays are described below (Singh.2015).

(1) Processing Delay (d_{proc})

Processing delay is the time taken by the router to access the header of a data packet and pass it to the next hop (Router). It also includes checking the next destination address as well as checking for any bit error that can occur during the transmission. Processing delay of a high speed router is in microseconds; however it takes a high processing delay due to processing of encryption and decryption algorithms.

(2) Queuing Delay (d_{queue})

Queuing delay is the time that the packet has to wait in the queue of a router until it can be transmitted over the link. Normally a packet can be put in the waiting queue when the speed of incoming to the router is faster than the outgoing link. The queuing delay can be higher if the size of the queue (buffer) is very small.

(3) Transmission Delay (d_{trans})

Transmission delay is the time that taken to push all the packet's bits on to the link and is usually caused by the data rate of the link. It is given by the following formula,

$D_T = N/R$,

D_T = The transmission delay, N = Number of bits and R = Rate of transmission

For example, if the data rate of the link is 10Mbps and the packet size is 100Kbps. Then the transmission delay can be calculated as 100Kbps/10Mbps = (100×1000)/(10× 1000000) = 1/100 seconds = 0.01 of a second.

(4) Propagation Delay (d_{prop})

Propagation delay is the time that taken by the 1st bit (a signal) of the packet to travel from source to the destination. It can be calculated by dividing the distance between the two routers and the speed of propagation of the link.

D = distance between the two routers

S = Speed of prorogation

Propagation Delay = D/S

(5) Nodal Delay (d_{nodal}) = (d_{queue}) + (d_{proc}) + (d_{trans}) + (d_{prop})

Delay of a data packet refers as Nodal Delay which described in the above equation. The perception of network speed and performance is usually understood as bandwidth, and latency is the only key element.

The bandwidth is the maximum data rate that is supported by a network connection or an interface and it represents the capacity of the connection which the internet provider promised. The greater the capacity, and more likely that greater performance will follow. However, the overall performance will depend on several factors such as latency and actual throughput. Network throughput is the actual speed at which data can be transferred to a device, network throughput can be reduced due to several factors including network latency and limitation of software/hardware that form a part of the network.

However, we need to understand the distinction between theoretical throughput and real-world results. As an example, a 1000 BASE-T (unshielded twisted-pair cables) Gigabit Ethernet (GbE) can theoretically support 1000 megabits per second (Mbit/s), but this level in practice can never achieved due to many factors including hardware and system overhead. (Prathap, 2015). However, the bandwidth cannot exceed the network hardware capacity.

As described earlier, there are several factors that affect total VM migration time and the VM downtime. Mainly, VM load and the bandwidth are the two most important factors that affect VM migration time. VM memory migration method affects the migration downtime, but

current VM memory migration techniques have matured over the years and reduce VM downtime significantly.

Figure 6.4 shows a significant reduction of VM migration latency and VM downtime when increasing network bandwidth. According to the Gilad Shainer, typical cloud data centres across the globe support 10 Gbps network bandwidth, and they are planning to move to 40 Gbps by late 2019 (Shainer, 2014) . We assume that a VM that has 5 GB of memory needs to migrate from host A to Host B on a 10 Gbps link, and it takes only 4 seconds (5GB/10 Gbps) for migration.

6.2.3.1 Calculating Required Bandwidth for VM Migration by Setting Up a VM Migration Threshold.

Calculating required bandwidth for VM migration is a function of the Bandwidth Predictor component of the framework, and it then passes that calculation to the Load balancer. The proposed VM placement algorithm tries to reduce network bottlenecks by calculating required bandwidth prior to VM migration and providing minimum bandwidth for VM migration. Setting up a VM migration window through a VM migration time threshold can be used to calculate the required bandwidth for migrating VMs

The threshold value can be set up by the administrator according to the requirements of the data centre. The idea behind setting up a VM migration time threshold is trying to complete the VM migration in less than or equal to the VM migration time threshold. The shorter the VM migration time the better, because when a VM is running a live service, it is important that the VM migration occurs in a manner that balances the requirements of minimising both downtime and total migration time. And also it's important to reduce the total migration time, because if VM migration traffic occupies the network for longer period, then it will create network bottlenecks.

Once the VM selection algorithm selects the VMs for migration, the following formula calculates the required bandwidth for the migrating VMs.

$$RequiredBandwidth = \frac{VMmemory\ Size}{VM\ Migration\ Time\ Treshold}$$

As an example, we assume that a VM1 with 2 GB virtual memory is moving from server A to B, and the maximum network bandwidth is 1 Gbps. We assume that the available bandwidth

of the network is 500 Mbps due to other traffic, and we assign the VM migration time threshold to 10 seconds (Figure 6.5).

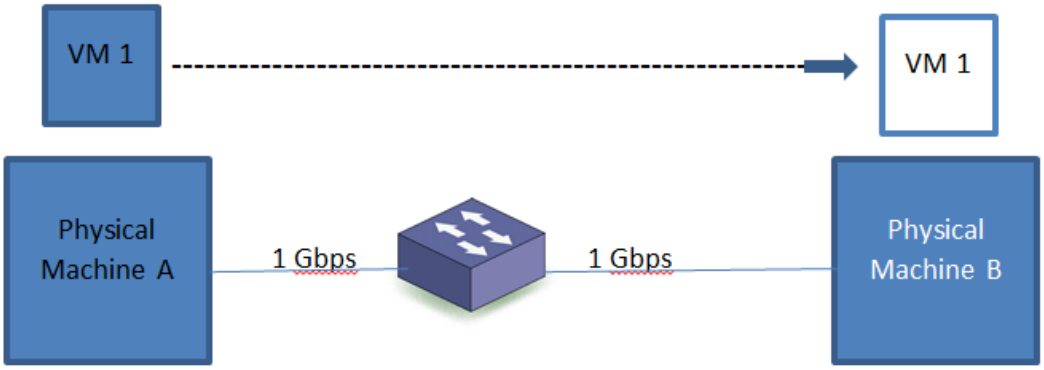


Figure 6.5: Required Bandwidth Calculation for VM Migration

The required bandwidth for VM migration will be.

$$\text{Required Bandwidth} = \frac{500\ \text{MB}}{10\ \text{Seconds}}$$

$$= 50\ \text{MBps}$$

To calculate Megabytes per seconds (MBps) to Megabits per seconds (Mbps), the final value must be multiplied by 8.

$$\text{Required Bandwidth} = 50\ \text{MBps} \times 8$$

$$= 400\ \text{Mbps}$$

6.2.3.2 Selecting a Best Host for Migrating VMs

The next problem is selecting a new destination PM for a migrating VM. What parameters should be considered while selecting a new host for a migrating VM? As we discussed earlier, the VM placement seen as a bin packing problem with variable bins where bins represent hosts and the bin size are available resources (CPU, memory, bandwidth and etc) of the host. Existing hosts selecting approaches select a PM which is least loaded, most approaches used host available CPU capacity and the available host RAM. But what would happen if the host doesn't have enough bandwidth to support VM migration? Therefore, in the proposed VM placement algorithm considers available CPU capacity, RAM and available bandwidth of the host are considered.

6.2.3.3 Prioritising VM Migration Traffic by Controlling User Traffic on the Network

The proposed framework is designed to provide minimum bandwidth for migrating VMs even if there is not enough bandwidth on the network to support VM migration traffic. As it is discussed in the literature review, VM migration traffic (VM memory size and swapped out page size) and several different types of network traffic (user traffic, VM management traffic, shared storage traffic and etc) on the network would compete for available bandwidth. There are different types of applications running on VMs and they generate a huge amount user traffic in a cloud data centre.

For simplification, this research only focuses on VM migration traffic and user traffic. In VM migration traffic, only the VM memory size is considered due to simplifying the implementation of algorithms in CloudSim as no option is offered to replicate VM swapped out pages. In addition, the CloudSim simulator doesn't offer an option to replicate user traffic either. However, in the CloudSim simulator, there is a class call Cloudlet, which is a collection of tasks or jobs. A Cloudlet can model the cloud-based application services such as content delivery, business flow, social networking and etc (Calheiros et al.2011). In CloudSim, it sends Cloudlet to VMs randomly to process, and the process consumes network bandwidth, so we presume that a Cloudlet is a collection of jobs sent by users and VM bandwidth utilization occurs due to user's engagement with VM applications. In the simulation, we presume that the VM bandwidth utilisation occurs due to user traffic.

As VM migration traffic consumes more bandwidth in a very short time period, the VM migration traffic must be given more priority over user traffic and other network traffic on the network. If there is not enough bandwidth on the network to support VM migration, the proposed VM placement bandwidth provides enough bandwidth by controlling user traffic on a VM which has the least user traffic.

Detection of Underutilised VMs

In cloud-based data centres, a different level of bandwidth is assigned per VM according to customer requirements. We can calculate the VM bandwidth utilisation percentage from the assigned bandwidth using the formula which is shown in the Eq 6.5, and we assume that the VM bandwidth utilisation is due to user traffic to simply the simulation. Then we set up a VM bandwidth utilisation threshold (30 % from assigned bandwidth) to detect least

bandwidth utilised VMs. The bandwidth utilisations of all VMs that are under 30 % of assigned bandwidth are considered as underutilised VMs. Proposed VM placement algorithms stop user traffic on underutilised VMs for the amount of time that equals to the VM migration threshold to provide enough bandwidth for the VM migration.

$$\textit{VM Bandwidth Utilisation \%} = \frac{VM\ \ Bandwidth\ \ Utilisation}{VM\ Assinged\ Bandwidth} \times 100 \qquad \textit{Eq (6.5)}$$

The Network Traffic Controller component is responsible for calculating the user traffic and controlling user traffic on VMs. For simulation purposes, the functions of the Bandwidth Predictor, Network Traffic Controller, and Global VM Migration Manager components have been implemented in the proposed VM Placement algorithm.

Proposed Bandwidth Guaranteed VM Placement Algorithm

Input*: hostList,VMlist, network bandwidth capacity , PmAvailableBandwidth , SelectedVm List* (selected for migration) , *UnderloadedPmList , VmUUIDs* (universal unique identifier), *VMmigrationTime Threshold , assigned bandwidth of each VM , Available Bandwidth, Total Network Bandwidth*

Output: *VM Placement*

1. ***Step (a) Potential hosts for migrating VMs***
2. *Get the UnderloadedPmList from the database*
3. *Sort out the UnderloadedPmList in Ascending order according to memory utilisation*
4. ***Step (b) Potential VMs that are selected for migration***
5. *Get the SelectedVmList from the database (VMs ready to migrate)*
6. *Sort out the SelectedVmList in ascending order according to the VM memory size*
7. *Map the SelectedVmList to their VmUUIDs*
8. ***Step(c) Calculate the load of each VM that needs to be migrated***
9. *Get the SelectedVmList from the database*
10. *Get the VM memory size of each VM in the SelectedVmList*
11. *Sort out VMs in ascending order according to the VM memory size*
12. ***Step (d) Network bandwidth & user traffic aware VM placement***
13. *Get the host memory size of each PM in UnderloadedPmList from the database*
14. *Get the memory size of each VM in the SelectedVm List*
15. *Get the bandwidth utilisation data of all PMs from the database*
16. *Declaring VMmigrationTime Threshold = 10 seconds (migration window keeps VM migration time under 10seconds)*
17. *Get the available bandwidth (AveblBdw)on the network*
18. *Get the total network bandwidth*
19. ***for*** *vm in SelectedVmList* ***do***

20. *calculate the required bandwidth for each migrating VM*
21. *RequiredBandwidth* = $\frac{VMmemory\ Size}{VMmigrationTime\ Threshold\ \ (10\ seconds)}$
22. ***If*** *RequiredBandwidth for migrating VM* $\leq$ *AveblBdw* ***and***
23. *hostRam in the UnderloadedPmList* $\geq VmRam$ *of migrating VM* ***then***
24. *mapping the vmUUID to the hosts in the UnderloadedPmList*
25. *Mapped* $\longleftarrow$ ***true***
26. *Allocate Vm to the host* $\longleftarrow$***true***
27. *Start the migration with the VMs that has less memory first*
28. ***else***
29. *Get the bandwidth utilisation data (VMbdwUsge) for each VM*
30. *Get the assigned bandwidth of each VM from the database*
31. ***for*** *VM in the VMlist* ***do***
32. *Calculate the VM bandwidth utilisation percentage from the assigned bandwidth for each VM (eq6.4)*
33. ***If*** *any VM's bandwidth utilisation is* $\leq$ *30 %* ***then*** *mark them as underutilisedVMs*
34. ***end if*** *RequiredBandwidth for migrating VM* $\geq$ *AveblBdw* ***then***
35. *stop sending Cloudlets (similar to stopping user traffic) to underutilisedVMs for the duration of == VMmigrationTime Threshold (10 seconds)*
36. *mapping the vmUUID to the hosts in the UnderloadedPmList*
37. *Mapped* $\longleftarrow$ ***true***
38. *Allocate Vm to the host* $\longleftarrow$***true***
39. *Start the migration with the VMs that has less memory first*
40. ***end if***
41. ***end for***
42. ***return*** *VM placement*

6.4 Comparison and Performance Analysis

As it was stated in the literature review, cloud datacenters are equipped with heterogeneous physical machines (PMs) and VMs and the characteristics of PMs and VMs change according to the requirements of the data centre. Workloads running in these data centres typically consist of diverse applications with different objectives and resources requirements. The proposed framework is based on four main algorithms, which are; underloaded host detection, overloaded host detection, VM selection and VM placement algorithms. These algorithms reside in the Load balancer component in the framework. The proposed four algorithms will be implemented in the simulation scenario which is presented in the Experimental Setup section.

6.4.1 Benchmark for Evaluating Proposed Algorithms in the Framework

Due to the necessity of carrying out the simulation in a large-scale infrastructure, a CloudSim simulation tool has been considered in this research, because it provides support for modelling and simulation of virtualised cloud based data centre environments including dedicated management interfaces for VMs, memory, storage, and bandwidth. As a benchmark for the evaluation, two generic VM selection and allocation policies in CloudSim have been considered to evaluate the proposed algorithm, the benchmark policies are:

(1) Static Threshold (Thr) VM Allocation and Random Selection (RS) Allocation Policy

This VM allocation and selection policy contains four algorithms, which are host underloaded, host overloaded, VM selection and VM placement algorithms which are described below.

Host overloaded – When PM's CPU utilisation reaches over 80 % from the total capacity.

Host Underloaded – All PMs CPU utilisation is less than 30 % from the total capacity.

VM selection – It selects VMs for migration randomly from overloaded hosts.

VM Placement – It places VMs on underloaded hosts randomly.

(2) Static Threshold (Thr) VM allocation and Minimum Migration Time (MMT) VM Selection Policy (ThrMmt)

This VM allocation and selection policy contains four algorithms, which are host underloaded, host overloaded, VM selection and VM placement algorithms which are described below:

Host overloaded – When PM's CPU utilisation reaches over 80 % from the total capacity.

Host Underloaded – Selects PMs CPU utilisation less than 30 % from the total capacity.

VM selection – It selects VMs which have the minimum amount of RAM for migration in order to reduce the VM migration time.

VM Placement – It places VMs on underloaded hosts random.

CloudSim Policy Name	Convention	Resources Type
Static Threshold VM Allocation and Random Selection	ThrRs	Host CPU utilisation
Static Threshold VM Allocation and Minimum Migration Time VM Selection	ThrMMT	Host CPU utilisation, and VM memory

Table 6.1*:* Benchmark Algorithms in CloudSim

6.4.2 Experimental Setup

For comparison, the proposed algorithms, an IaaS environment has been considered and it represents a large-scale cloud datacenter consisting with *m* heterogeneous physical nodes (Figure 6.6). The Central Controller resides on the master node, the Local Controller is responsible for collecting data from PMs and VMs and storing it in local file-based storage, then it periodically replicates data with the central database. The Central Controller fetches data from the central database and then it issues VM migration commands to balance the load of the system.

In this simulation, proposed algorithms have been implemented in CloudSim 3.0.3 to analyse the performance of the proposed algorithms. In the simulation, it created one data centre, and the simulation considered 50 heterogeneous physical nodes. The halves of physical nodes are HP ProLiant G4 and HP ProLiant G4. Two VMs per host have different types of configurations making 100 total virtual machines in the data centre. The Xen hypervisor (VMM) has been used, and also a total of 1700 cloudlets (tasks) have been used in the simulation and randomly assigned tasks to VMs. The simulation has run for a virtual 24 hours (actual 5 hours) to evaluate the performance of the proposed algorithm. The configurations of PMs, VMs, and the Cloudlets are shown in the table 6.2, 6.3, and 6.4.

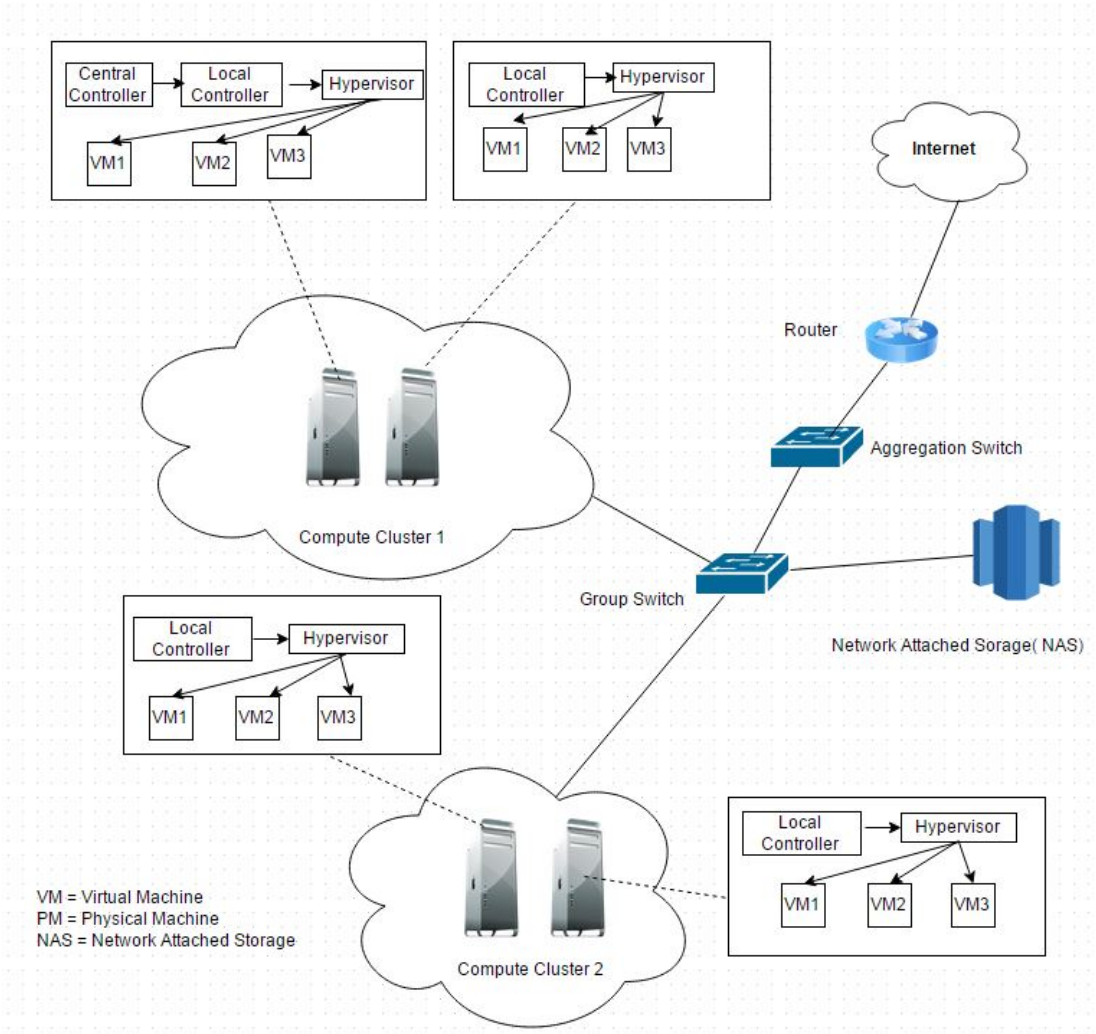


Figure 6.6: System Model

Physical Machine Configuration (Host)

Brand	CPU Cores	Core 1 MIPS	Core 2 MIPS	Network Bandwidth	Memory	Storage	Operating System	Total Hosts
HP ProLiant G5	2	2600	2600	1.2 Gbps	4.096 GB	2 GB	Linux x86	25
HP ProLiant G4	2	1860	1860	1.2 Gbps	2 GB	1 GB	Linux x86	25

Table 6.2: Physical Machine Configurations

VM Configuration

VM Types	CPU Cores	Core 1 MIPS	Memory	Storage	Network Bandwidth	Operating System	Total Machines
VM Type 1	1	1000	512 GB	2.5 GB	512 Mbps	Linux x86	50
VM Type 2	1	1600	1.5 GB	2.5 GB	512 Mbps	Linux x86	50

Table 6.3: VM Configurations

Cloudlet Configuration

	Cloudlet Length in Million Instructions (MI)	Long File Size in Bytes	Long Output Size in Bytes	Simulation Length	Total Cloudlets Used
Cloudlet	300*60*60*24	300	300	1 day (60 sec * 60 minutes * 24 hours)	1700

Table 6.4: Cloudlet Configuration

6.4.3 Evaluation Metrics

The performance of the framework is based on the combination of four algorithms which reside in the Load balancer component. Those four algorithms work dynamically, the main priorities of the framework are to balance the load and reduce network bottlenecks by prioritising VM migration traffic over user traffic. Therefore, combinations of five evaluation metrics have been considered in order evaluate the performance of the proposed algorithms in the framework.

Number of SLA Violations

Meeting QoS (Quality of Service) is extremely important for the Cloud computing environment, normally QoS requirements of the customer are formalised in the form of SLAs. Service Level agreement is kind of agreement between the customer and the cloud provider regarding quality, priorities, and responsibilities. These SLAs can be determined in terms of characteristics such as minimum throughput, completion time, maximum response time, availability, minimum bandwidth and latency. These characteristics are workload or

application dependent, from the perspective of Cloud providers, so it's very important to fulfil SLA requirements to keep their customers happy.

When a VM cannot get the promised QoS, SLA violation occurs. To avoid SLA violation, VMs were packed on the host such in a way that the host utilisation was kept below pre-defined CPU, memory and bandwidth thresholds.

When a VM migration fails to complete its process within its pre-specified window, it can degrade the QoS experienced by the affected VMs and it may cause number of SLA violations due to the following reasons.

- A VM cannot get requested resources (MIPS, memory or bandwidth) SLA violation occurs.
- The duration of the VM migration can last lot more than estimated due to network bandwidth shortage.
- VMs will not get allocated bandwidth due to network bandwidth shortage.

It's necessary to define workload independent metrics that can be used to evaluate the QoS delivered for any VM deployed on the IaaS (Infrastructure as a Service). Anton Beloglazov proposed a metric called the overall performance degradation due to VM migration (PDM) (Beloglazov et al.2011). It calculates the CPU performance degradation of all VMs in the data centre due to VM migration by taking the VM's CPU capacity and the estimated CPU performance degradation into consideration. If the VM doesn't get the requested CPU capacity (MIPS) during a VM migration, SLA violation occurs. Performance Degradation due to Migration (PDM) SLA violation metric is shown in Eq 6.5.

$$PDM = \frac{1}{M} = \frac{C_{d_j}}{C_{r_j}} \qquad \text{(Eq 6.6)}$$

M is the number of VMs in the data centre; C_{d_j} is the estimate of the performance degradation of the VM j caused by VM migrations; C_{r_j} is the total CPU capacity requested by the VM j during its life time. In this simulation, C_{d_j} is estimated to be 10 % of the CPU utilisation in MIPS of all migrations of the VM j.

Based on the literature review and previous work, most SLA violation metrics measured performance degradation due to a VM migration by taking CPU capacity and estimated CPU performance degradation into consideration. As this research focuses on network bandwidth,

it needs an independent SLA violation metric to evaluate the performance of the proposed algorithms by taking the bandwidth degradation of VMs due to VM migrations into consideration. In a typical Cloud data centre, each VM has an assigned bandwidth according to its user requirements, and Cloud providers must make sure that those VMs get the allocated bandwidth and guarantee to never have less capacity than they were individually assigned.

VM migration can put an extra strain on the data centre network, and sometimes this can cause bandwidth degradation on VMs in the data centre if there is not enough bandwidth to support the migration. In consequence unnecessary VM migrations can cause SLA violation, but taking only the PDM SLA violation metric can't evaluate the bandwidth degradation due to the VM migration. Therefore, a Bandwidth Degradation due to VM Migration (BDM) metric has been proposed. The reason behind the BDM metric is the observation that the VM is getting 100 % of the assigned bandwidth during a VM migration. The proposed BDM SLA metric is shown in the Eq 6.6.

$$BDM = \frac{1}{N} \sum_{j=1}^{N} \frac{B_{dj}}{B_{aj}} \qquad \text{(Eq 6.6)}$$

Where N is the number of VMs; B_{aj} is the total bandwidth (mbps) assigned to *VM j* during its lifetime. This work assumed that B_{dj} is estimated bandwidth degradation, which is 10 % of the total assigned bandwidth in mbps (megabits per second) during all migrations of the *VM j*, and j being the active host (serving VMs). Both PDM and BDM metrics independently measure SLA violations in the system.

Number of VM Migrations

Live VM migration causes significant performance degradation due to VM downtime during the VM migration process. Moreover, multiple simultaneous VM migration puts extra strain on the underlying infrastructure, and migrating VMs will stress the cloud resources. Too many VM migrations may achieve balanced loads, but lead to performance degradation. This is an auxiliary metric that represents the performance and it is measured with other metrics together (Xu et al. 2010). Therefore, the proposed algorithm will try to reduce the number of VM migrations, noting that less VM migration shows better performance.

Number of Overloaded Hosts

This metrics measure how many hosts in the cluster are overloaded which gives an overview of the system status. The value depends on the reconfiguration of the overloaded threshold. The Load balancing algorithm aims to reduce the number of overloaded hosts as much as possible, because if more overloaded hosts are detected, some of its VMs need moving to other hosts to balance the load of the host. Unnecessary overload detection would cause needless VM migration which may cause performance degradation. This is a straightforward metric to evaluate a load balancing effect (Xu et al.2017). The proposed framework will try to reduce the number of overloaded machines in the cluster.

Number of Underloaded Hosts

VM consolidation offers significant benefits to cloud data centres, energy efficiency in a data centre can be achieved through dynamic VM consolidation. Energy efficiency typically can be obtained by transitioning idle physical machines into a power saving state during periods of low utilisation. To facilitate the creation of idle physical machines, dynamic VM consolidation can be achieved through VM live migration. The objective of dynamic VM consolidation is to pack the VMs on the least number of PMs dynamically while minimizing the number of VM migrations. An increased number of underloaded hosts in a data centre shows better VM consolidation by moving VMs to a minimal number of hosts in the data centre.

Energy Consumption

Energy consumption is a very important factor in a cloud data centre, the power and energy consumed by the physical machines, network devices, cooling systems and other computer equipment are major factors that contribute to high energy cost, and high carbon emission. In addition, VM migration consumes energy, according to an experiment by Dhanoa et al, energy consumption of VM migration depends on the VM size and available network bandwidth. VM size has a linear relationship with energy consumption and migration time (Dhanoa et al.2015). VM size and the available network bandwidth affect migration time and energy consumption of the host physical machine.

Previous study has shown that CPU utilisation has a linear relationship with the power consumption (Chen et al.2015). According to the Beloglazov et al, in average an idle server consumes 70 % of the power consumed by the server running at full CPU speed (Beloglazov

et al.2010). This fact justifies that a data centre can save energy by shutting down idle servers. Beloglazov et al proposed a power consumption model which calculates energy consumption of a PM by taking the host CPU consumption into consideration (Beloglazov et al.2010). The host CPU power model is defined in eq 6.7.

$$P(u) = k \times P_{max} + (1 - K) \times P_{max} \times u \quad \text{(Eq 6.7)}$$

Where P_{max} is the maximum power consumed when the server is fully utilised, k is the fraction of power consumed by the idle server, and u is the host CPU utilisation. The CPU utilisation of the host may change over time due to variability of workloads. The CPU utilisation is a function of time and represented by $u(t)$. Therefore, the total energy (E) consumption by a physical machine can be defined as an integral of the power consumption function over a period of time, which is defined in eq 6.8.

$$E = \int P(u(t)) \qquad \text{(Eq 6.8)}$$

Eq 6.7 and 6.8 have been implemented in the simulation to calculate the energy consumption in the data centre, and the final value of the eq 6.8 must be multiplied by 50 as the simulation used 50 physical machines.

6.4.4 Simulation Results and Analysis

The simulation scenario has been implemented on a testbed in CloudSim 3.0.3, and 1700 Cloudlets (tasks) have been used in the simulation, the Cloudlet simulation length was 24 hours and it ran for a virtual day in CloudSim (actual 5 hours). In order to compare the efficiency of the proposed VM allocation and VM placement policy, we compared it with two policies that are already built into the CloudSim, which are ThrRs, and ThrMMT based on five evaluation metrics (SLA violation, number of overloaded and underloaded hosts, energy consumption, and number of VM migrations). The results are shown in the table 6.5 and 6.6, figure 6.7, 6.8, 6.9, 6.10, 6.11, 6.12 and 6.13.

Name of the VM Allocation Policy	SLA Violation % (PDM)	SLA Violation % (BDM)	Number of Overloaded Hosts	Number of VM Migrations	Energy Consumption KWh	Number of Underloaded Hosts	Number of Successfully Completed Tasks
ThrRs	0.12	0.17	195	3315	30.36	61	694
ThrMMT	0.09	0.13	194	2474	26.77	62	695
Proposed	0.04	0.05	130	2066	18.05	98	696

Table 6.5: Comparison Results of ThrRs, ThrMMT and Proposed Policies According to Evaluation Metrics.

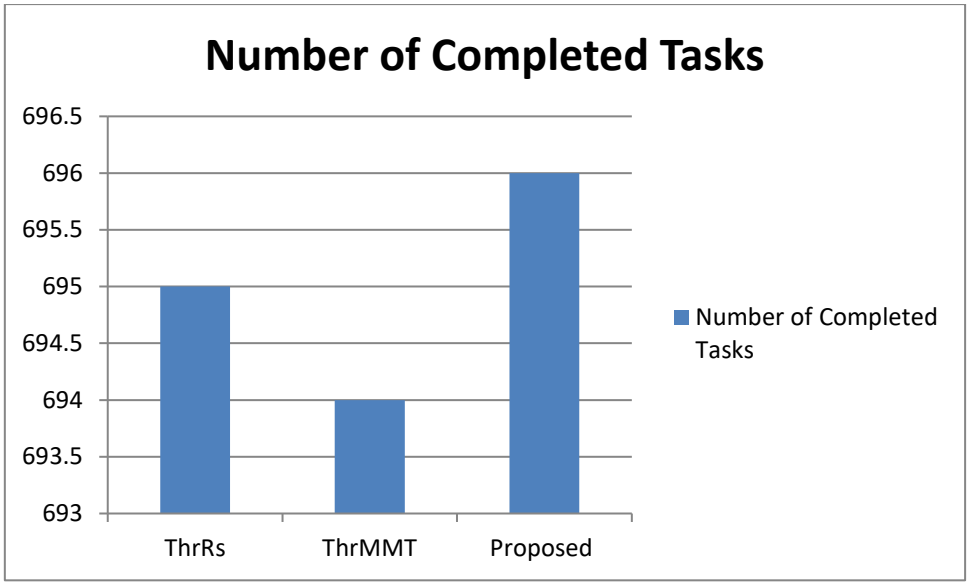


Figure 6.7: Shows Number of Successfully Completed Tasks

6.4.4.1 Number of Overloaded Hosts

According to the result presented in the figure 6.8, it shows that the proposed VM allocation and placement policy significantly reduced the number of overloaded hosts when compared to ThrRs and ThrMMT benchmark policies. The proposed policy has recorded nearly 65 less overloaded machines than the ThrRs policy while 64 less overloaded machines recorded

than ThrMMT policy. ThrRs and ThrMMT policies recorded almost similarly overloaded hosts, because both policies use the same upper and lower thresholds (80% and 30%) and host CPU utilisation to detect overloaded hosts. Figure 6.8 shows the number of overloaded hosts that have been recorded in the simulation.

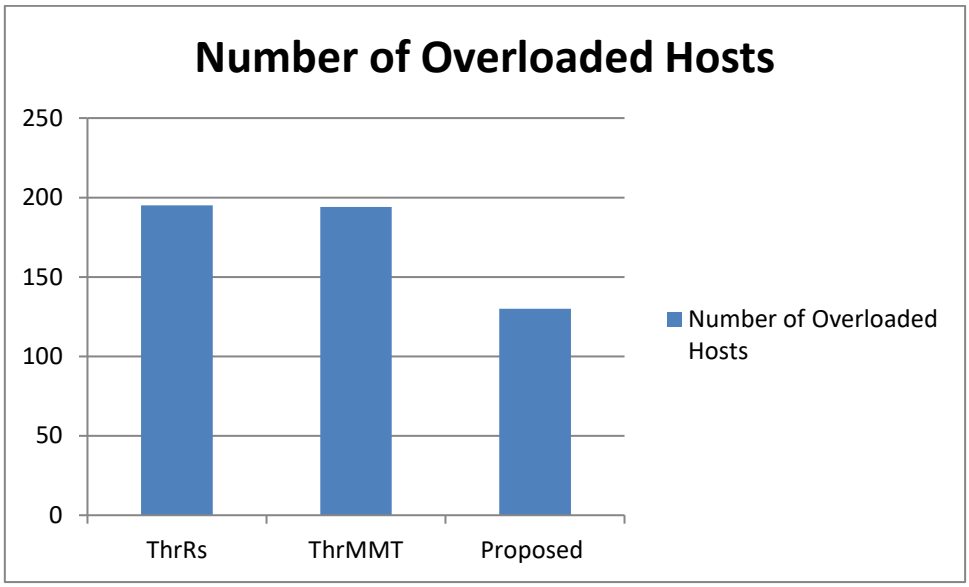


Figure 6.8: Number of Overloaded Hosts in the Simulation Results

The ThrRs and ThrMMT, both benchmark VM allocation and placement policies are considered as only single resource in their overloaded detection algorithm, which is the CPU utilisation of the host. Benchmark policies recorded more overloaded hosts, because a small spike of the host's CPU utilisations might have created a false alarm and unnecessarily created overloaded machines.

Due to that reason, taking only the host CPU utilisation for detection of overloaded machines will not give correct status of the system. In order to ensure that a small transient spike does not trigger needless overloaded detection and VM migration, multiple resources have been considered in the proposed algorithm. In the proposed overloaded detection algorithms, the host's CPU, memory and bandwidth are taken into consideration for overload detection and figure 6.8 shows that the proposed algorithm out-performed the benchmark policies with respect to the number of overloaded machines.

6.4.4.2 Number of Underloaded Hosts

According to the result presented in the figure 6.9, it shows that the proposed VM allocation and placement policy significantly increased number of underloaded hosts when compared with the ThrRs and ThrMMT policies.

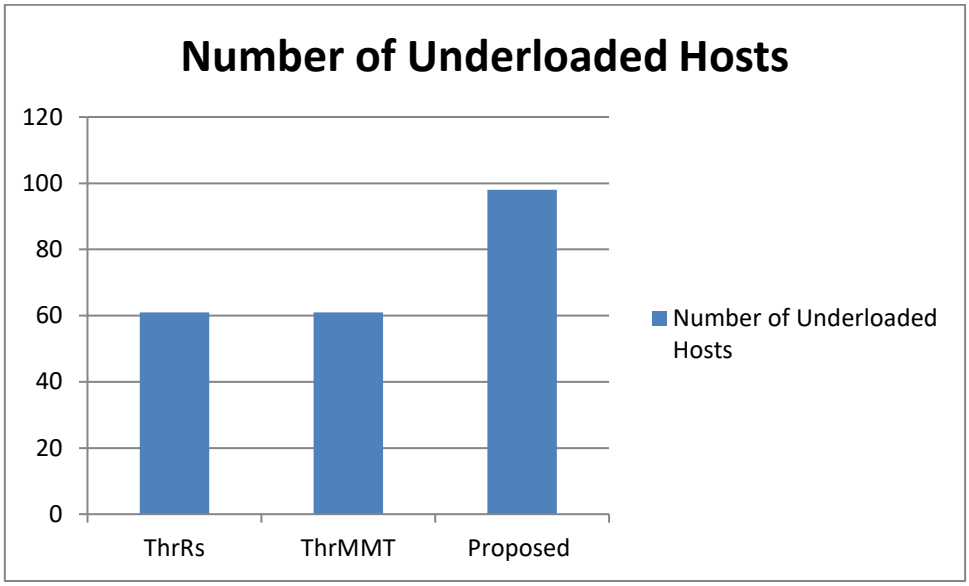


Figure 6.9: Number of underloaded machines in the simulation results

Both benchmark policies (ThrRs, ThrMMT) have recorded almost similar underloaded hosts (61, 62) in the simulation results while the proposed policy recorded 98 underloaded hosts, which is a significant increase when comparing with both benchmark policies.

VM consolidation offers significant benefits to cloud data centres. Dynamic VM consolidation is the first step towards achieving energy efficiency in a data centre. Energy efficiency typically can be obtained by transitioning idle physical machines into a power saving state during periods of low utilisation. To facilitate the creation of idle physical machines, dynamic VM consolidation can be achieved through dynamic VM live migration. The objective of dynamic VM consolidation is to pack the VMs on the least number of PMs dynamically while minimizing the number of VM migrations. Then idle severs can be switched off to save energy in a data centre. The simulation results show that the proposed policy of an increased number of underloaded hosts in the data centre, which shows better VM consolidation can be achieved by moving VMs to a minimal number of hosts in the data

centre. Additionally, the results indicate that VM consolidation is an effective way to improve the utilisation of the data centre's resources and QoS (Quality of Service).

6.4.4.3 SLA Violations and Number of VM Migrations

A high number of VM migrations has a negative impact on the SLA violation rates and there is a relationship between two metrics. Therefore, we analysed the performance of two benchmark policies and the proposed policy with respect to a combination of two SLA violations (PDM and BDM) parameters and number of VM migrations. Figure 6.10 and 6.11 show the comparison results with respect to SLA violations due to VM migrations.

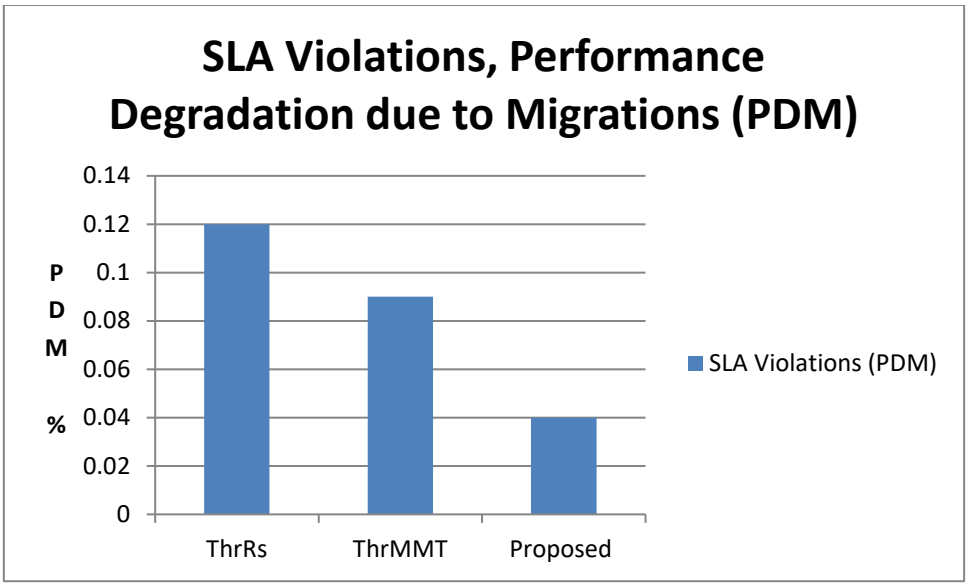


Figure 6.10: Comparison Results, SLA Violations (PDM)

Figure 6.10 shows the comparison results of SLA violation based on performance degradation due to VM migration. Figure 6.10 shows that ThrRs and the ThrMMT policies have recorded 0.12 % and 0.09 % SLA violations (PDM) respectively. The results show that ThrMMT has less violation levels comparing to ThrRs, because ThrMMT uses minimum migration time policy for VM migration. This reduces performance degradation resulting from VM migrations, because it selects VMs with minimum time (VM memory/bandwidth) required for VM migration. The proposed policy has significantly reduced SLA violations (PDM) and when compared with the two benchmark policies, the proposed policy recorded 0.04 %.

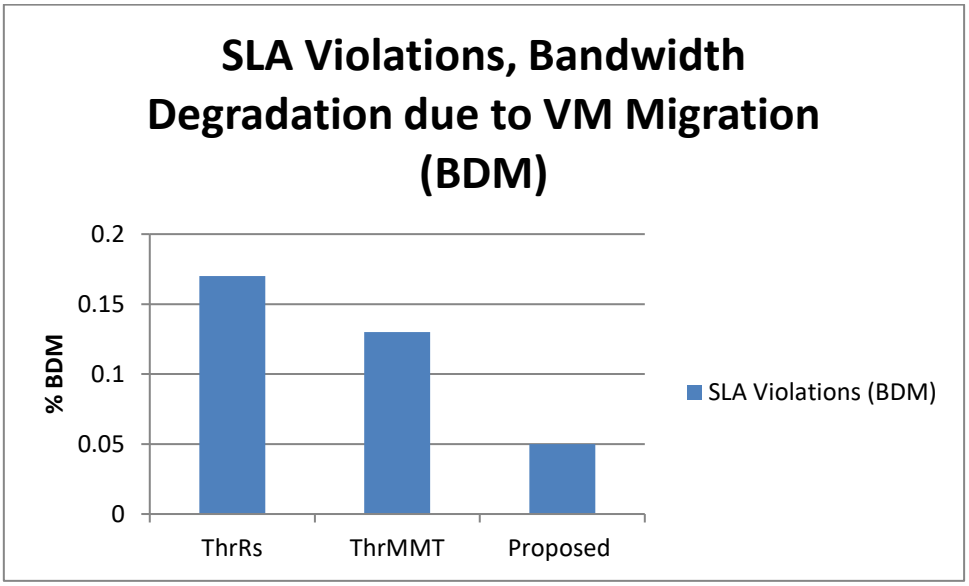


Figure 6.11: Comparison Results, SLA Violations (PDM)

Figure 6.11 shows the comparison results of SLA violation based on bandwidth degradation due to VM migration (BDM). Figure 6.11 shows that the ThrRs and ThrMMT policies have recorded 0.17 and 0.13 SLA violations (BDM) respectively. The results show that ThrMMT has lower violation levels compared to ThrRs policy, the difference being ThrMMT uses minimum migration time policy for VM migration. This reduces bandwidth degradation caused by VM migrations, because it selects VMs with minimum time (VM memory/bandwidth) required for VM migration but high SLA violations were still recorded. The proposed policy has significantly reduced SLA violations caused by VM migrations (0.05) when compared with the two benchmark policies.

Figures 6.10 and 6.11 show the simulation results of two SLA violation parameters, it indicates that the proposed policy has significantly reduced both the PDM and BDM SLA violation levels. The main reason for these significant reductions is that the proposed policy calculates the bandwidth required for migrating VMs prior to the actual migration by implementing a VM migration window (10 seconds in the simulation). If the data centre network hasn't got enough bandwidth to support the migration during that window, the framework will allocate enough bandwidth by stopping the user traffic (equals to the VM migration Window 10 seconds) on underutilized VMs (under 30 % utilisation of assigned bandwidth). For this reason, the proposed policy has outperformed the two benchmark

policies respective to the SLA violations (BDM and PDM) and significantly reduced SLA violation levels.

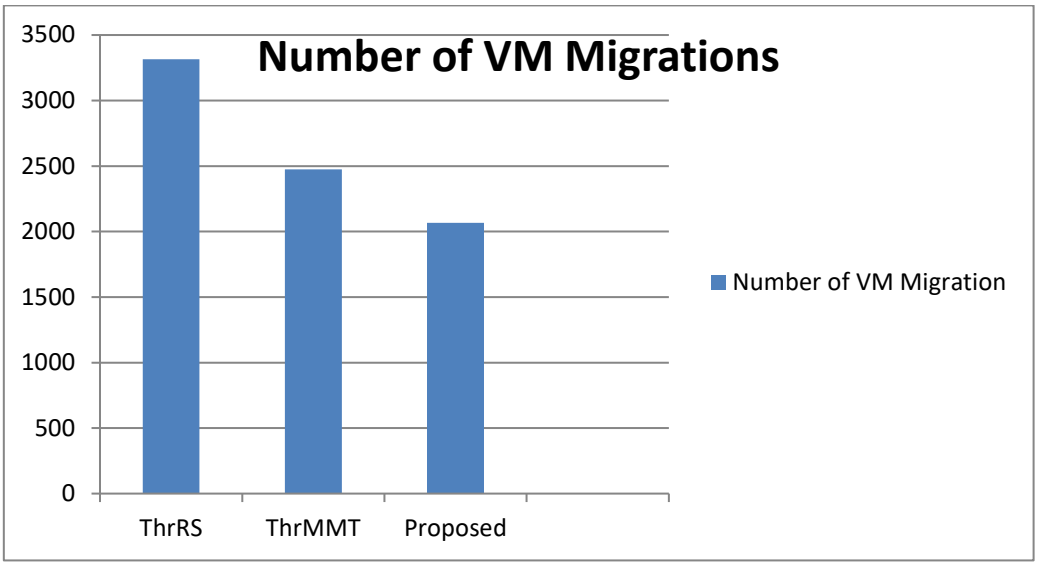


Figure 6.12: Comparison Results, Number of VM Migration

Figure 6.12 shows the results of a comparison between ThrRs, ThrMMT and the proposed policies with respect to the number of VM migrations. According to these results, ThrRs and ThrMMT have recorded 3315 and 2474 VM migrations respectively while the proposed policy recorded 2066 VM migrations. The proposed policy has shown a significant reduction of VM migrations compare to ThrRs policy, a total of 1249 migrations less than the ThrRs policy. The proposed policy also recorded 408 less VM migrations than the ThrMMT policy.

The ThrRs policy uses a random VM placement policy and doesn't take VM memory or other parameters into consideration when placing VMs, for this reason it recorded a high number of VM migrations. ThrMMT policy considers VM machine memory size when selecting the VM for migrations. It selects the VMs with minimum memory in order to minimize the total VM migration time. By doing that it managed to reduce 814 VM migrations compare to ThrRs policy.

The proposed policy considers CPU, Memory and Bandwidth utilisation when selecting overloaded and underloaded hosts and it increased accuracy in identification of overloaded hosts. This enabled it to significantly reduce the number of VM migrations noting that VM migration only takes place on the PMs which are more accurately identified as overloaded

hosts. Another reason is optimisation of the process of finding appropriate hosts for migrating VMs.

The proposed VM selection and allocation policy considers maximum CPU, minimum VM memory and maximum bandwidth in its VM selection policy. When identifying hosts for migrating VMs, it considers host memory (migrating VM's memory must be less than the host memory). Due to this reason, unnecessary VM migrations to inappropriate hosts are to a great extent eliminated. The proposed policy has therefore outperformed two benchmark policies respect to number of VM migrations.

6.4.4.4 Energy Consumption

In this experiment, the total energy consumption in the data centre is calculated by taking the host CPU utilisation into consideration. In order to make the simulation simple, this experiment did not consider the energy consumption by other data centre devices, such as switches, routers and etc. This simulation has evaluated the performance of two benchmark policies (ThrRs and ThrMMT) which are built into CloudSim, and also the proposed policy with respect to energy consumption.

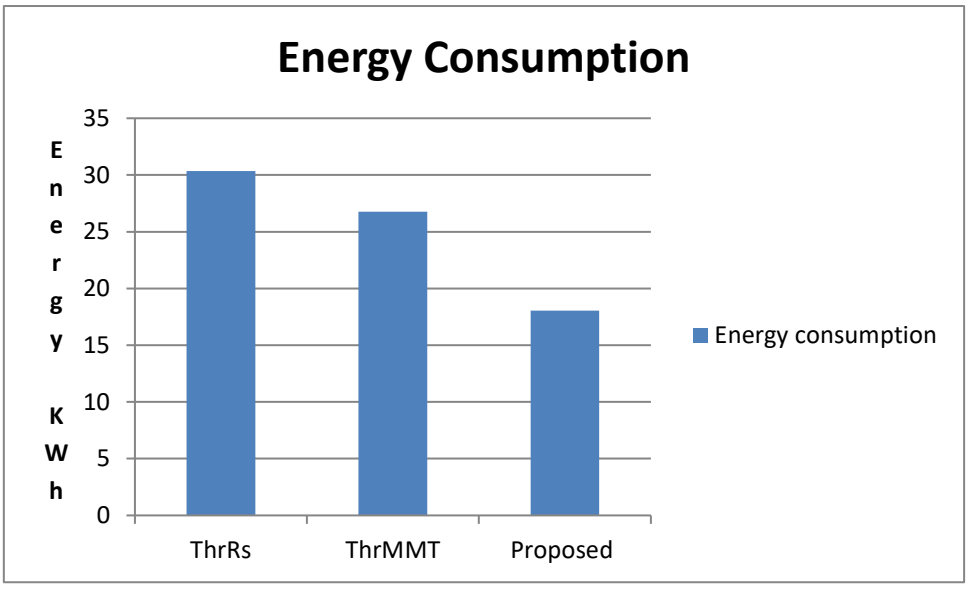


Figure 6.13: Comparison Results, Energy Consumption

In this experiment, energy consumption in the data centre is calculated by taking the host CPU utilisation into consideration as it has a linear relationship with power consumption.

One of the major issues in cloud data centres is a wastage of idle power when resources such as physical hosts providing computing and storage capacities run at low utilisation. For instance, a server which operates even at a very low CPU utilisation (10 %) consumes over 50 % of the peak power (Hameed et al.2014). According to the data provided by Intel Labs, the main part of power consumed by a server is drawn by the CPU (Beloglazov et al.2010).

According to Figure 6.13, ThrRs and ThrMMT policies recorded 30.36 KWh and 26.77 KWh respectively, while the proposed policy recorded just 18.5 KWh consumption. The simulation results indicate a significant energy reduction of the proposed policy when compared with the two benchmark policies. The reasons for reduction of energy in the data centre are:

- VM consolidation
- Efficient load balancing mechanism
- Reduction of VM migrations

VM consolidation reduced the number of active servers in the data centre by packing VM in minimal servers resulting in a reduction of energy consumption, because underutilized servers consume less energy than active servers, the load balancing mechanism in the proposed policy distributes the load among physical servers in an efficient manner, which utilise servers' resources efficiently including energy consumption. In addition, the proposed policy reduced number of VM migrations. Generally, the VM migration process consumes huge amount of energy and there is a reduction in the number of VM migrations in the proposed policy, which resulted in reduction of energy consumption. The results of the simulation indicate that the proposed policy outperformed the two benchmark policies with respect to energy consumption.

6.5 Chapter Summary

The implementation chapter has presented the design and implementation of the proposed framework. The proposed framework's main objective is prioritising VM migration traffic over user traffic thereby providing the required bandwidth for VM migration and reducing network bottlenecks on the network. The framework has two main components, the Central Controller and the Local Controller. The Local Controller collects statistical data from VMs and PMs and sends it to a central database. The Load Balancer component in the Central

Controller takes VM migration decisions based on the intelligence that gathered from its components. The components of the Central Controller are shown below:

- Global VM Manager
- Load Balancer
- Bandwidth Predictor
- Network Traffic Controller

The framework follows a centralised load balancing model, where the problem is divided into four main problems: host underloaded detection, host overloaded detection, VM selection and VM placement. Four novel algorithms have been proposed which are correspondent to each problem in the framework, and those algorithms reside in the Load Balancer component of the framework.

The four correspondent algorithms have been implemented in the CloudSim simulator, and two inbuilt benchmark policies (ThrRs and ThrMMT) applied in the CloudSim to evaluate the performance of the proposed algorithms (VM selection and placement policies). Five performance metrics have been used to compare and evaluate the performance of the proposed policy, which are: the number of VM migrations, number of overloaded hosts, number of underloaded hosts, energy consumption and SLA violations. Due to the lack of research and existing work relating to this research, it was difficult to find a SLA metric to measure the performance of the proposed policy in relation to bandwidth. Using the insight and knowledge that has been gained from a literature review, a new SLA metric, The Bandwidth Degradation due to VM Migration (BDM) has been proposed.

According to the simulation results (Table 6.5÷6.13), the proposed policy (a set of four algorithms) has shown to outperform the two benchmark policies (ThrRS and ThrMMT) with respect to five evaluation metrics (the number of overloaded hosts, number of underloaded hosts, SLA violation, the number of VM migrations and energy consumption). The proposed policy significantly reduced number of overloaded hosts, the number of VM migrations and SLA violations plus reduced energy consumption when compared with two benchmark policies. And also the proposed policy increased the number of underloaded hosts in comparison to the two benchmark policies.

Unnecessary and uncontrolled live VM migration will degrade performance and have a negative impact on the network, data centre resources, virtual machines and the SLA

violations. The key reasons for performance degradation due to uncontrolled and unnecessary VM migration are as follows: During a VM migration, each VM stops its operation for a very short amount of time in both VM memory transferring techniques (pre-copy and post-copy), this will cause problems for its users. During a live VM migration there is transfer of VM memory, CPU state and other parameters from the source to the destination host while the source's VM applications continue to run. The service quality of hosted applications is rigorously affected if VM migration is not properly handled. Live VM migration consumes a significant amount of network bandwidth as it transfers large-sized VM memory over a shared bandwidth, and too many VM migrations can put extra strain on the data centre's network. And also the resources on the source server are more utilised during a VM migration process, and overutilised servers will fail to provide required computing resources to its running VMs causing SLA violations. Most importantly, live VM migration causes a network bottleneck if there is not enough bandwidth to support the VM migration, and it will cause adverse effect on SLA violations and QoS. The simulation results indicate that the number of SLA violations rate increases with a high number of VM migrations.

The QoS requirements commonly formalised in the form of SLAs, which can be determined in terms of such characteristics as minimum throughput, maximum response time, bandwidth, and etc. The simulation results showed that an increasing number of VM migrations will have a direct impact on SLA violation levels and unnecessary VM migrations will cause more SLA violations and also negatively affect the QoS. Simulation results have shown that the two benchmark policies recorded a high number of VM migrations and SLA violation levels were high when compared with the proposed policy.

The obtained simulation results confirmed that the proposed VM selection and allocation policy significantly reduced SLA violation rates, because the proposed VM placement policy has a mechanism to calculate required bandwidth for migrating VMs prior to the migration by setting up a VM migration window (migration time threshold). If there is not enough bandwidth on the network, the framework is also able to provide the required minimum bandwidth for VM migration by controlling the user traffic on less utilised VMs in the data centre. In addition, the obtained simulation results confirmed that the proposed framework can be used as a dynamic load balancer, which uses live VM migration to balance the VM load on physical servers in a Cloud data centre with a limited impact on the VM performance. The results have also shown that the framework has achieved dynamic VM consolidation, which brings significant energy savings to a data centre.

Chapter 7: Conclusion

7.1 Summary

In recent years Cloud Computing has become more popular and it has made the vision of computing resources a utility and it offers utility-oriented IT services over the Internet to global users. As the technology improves and network speed becomes faster with lower latency, more users adopt the Cloud computing model. Users are utilising different types of applications from scientific to business providing cloud-based services in various forms, including software, hardware and data. Therefore, demand for Cloud data centres are expected to grow and accumulate a larger fraction of the world's computing resources. Large IT companies such as Google, Amazon, Microsoft, Amazon, IBM, and VMware have developed their own Cloud platforms and Cloud data centres to support the growing demand for Cloud based services.

As the Cloud is offering a variety of cloud based services, there are many different applications run on Cloud data centres making many companies and individuals rely on services through the Cloud. Due to the high demand, Cloud data centres are facing higher volumes of network traffic. According to Michael Kerner, Cloud traffic is also leading to an increased number of workloads per physical server. Cisco expected Cloud data centre traffic to grow to 14.1 ZB per year, with Cloud representing 92 % of the total traffic. As more companies and individuals rely on Cloud services, more strain is put on the data centre's resources. The most heavily affected limited resource in a data centre due to higher user demand is the bandwidth; giant Cloud data centres cannot deliver all their services to users if there is not enough bandwidth on the network.

As already mentioned, host virtualisation has matured and it has become a critical component of data centre and cluster computing operations. Host virtualisation offers live VM migration, which is a technique that migrates the entire system of a VM, including OS, memory, storage, process and network resources and also its associated applications from one physical machine

to another without disrupting the client or running applications. It's a key operation in current cloud installations and it can be used in flowing key scenarios.

1. Load balancing
2. Online maintenance and proactive fault tolerance
3. Power management

However, this research only focuses on live VM migration as a load balancer, which balances the VM load on physical hosts in a data centre.

One of the concerns with live VM migration its negative impact on data centre resources and active services. Live VM migration consumes a huge amount of network bandwidth, which depends on VM memory size and the available bandwidth on the network. Live VM migration is known to be an expensive operation because of the additional network traffic generated during a migration.

Live VM migration can cause network bottlenecks if there is not enough bandwidth to support VM migration, because both VM application traffic and VM migration traffic would compete for network bandwidth. Some virtualisation vendors currently recommend a separate network for VM mobility. However, setting up a separate network just for VM migration can be extremely costly and also it presents a barrier for seamless VM migration. Therefore, it is apparent that VM migration should be orchestrated in a network bandwidth-aware manner while being allocated the required bandwidth for VM migration .And also it is apparent that VM migration needs an intelligent, dynamic and automated mechanism to schedule and execute VM migration to prevent network bottlenecks.

To address the formulated research issues, this thesis has achieved each of its objectives that are presented in the Research Problems and Objectives section in the Chapter 1. Chapter 2 presented an in-depth review and analysis of Cloud computing.

Chapter 3 presented a taxonomy and survey of live VM migration, it reviewed and analysed in depth the live VM migration concept. Also it investigated different types of VM migration techniques, performance metrics, challenges of VM migration and live migration implementation in commercial Cloud platforms. Live VM migration can be used to address different issues in a Cloud data centre, and it offers many benefits to a Cloud data centre, which are discussed in Chapter 3. The proposed framework used live VM migration as a technique to balance the VM load on physical servers.

In Chapter 4, the dynamic load balancing based on VM migration, load balancing algorithms and existing implementations have been analysed. The research literature analysis has helped to identify gaps and open challenges in the research filed. Different types of load balancing algorithms and existing implementations have been analysed to obtain theoretical performance estimates and provide insight into designing proposed algorithms.

To address the next objective, Chapter 5 presented the proposed centralised framework for VM migration; the framework consists of two main components, the Central Controller and the Local Controller. The Central Controller has several components, the Load balancer, Bandwidth Predictor, Network Traffic Controller, and Global VM Migration Manager. The framework is relying on four proposed algorithms which reside in the Load balancer component and the algorithms gather information from other components that are needed in order to make VM migration decisions. The four main algorithms are: host overloaded detection, host underloaded detection, VM selection and VM placement algorithms. Moreover, the Central Controller of the framework will be replicated on another host and it can be used as a backup server eliminating a single point of failure.

Chapter 6 presented the implementation and evaluation of the proposed framework. The four algorithms of the framework have been implemented applying the CloudSim simulation tool, the functions of Bandwidth Predictor, Network Traffic Controller and Global VM Manager have been implemented in the VM placement algorithm. A collection of four proposed algorithms which is also called a VM migration policy in CloudSim. Two benchmark VM migration policies, ThrRs and ThrMMT have been used to compare the performance of the framework (four algorithms), which are inbuilt two policies in the CloudSim simulation tool. Five performance metrics have used to evaluate the performance which are the number of VM migrations, number of overloaded hosts, number of underloaded hosts, energy consumption and SLA violation. Due to the lack of SLA violation metrics to evaluate the VM bandwidth degradation caused by VM migration, a novel SLA violation metric, bandwidth degradation due to VM migration (BDM) has been proposed.

This thesis has investigated VM live migration, its existing techniques and related work, and then proposed a centralised framework for VM migration. The proposed framework improves data centre resource utilisation by balancing the VM load among physical servers in an efficient manner and achieving more efficiency in energy consumption through VM

consolidation. Most importantly, it reduced network bottleneck and SLA violations while satisfying the defined QoS requirements.

7.2 Future Research Directions

Despite a substantial contribution of this research in relation to bandwidth-aware live VM migration, there are number of areas that need to be improved in the framework. And also there are number of open research challenges that need to be addressed.

The proposed overloaded and underloaded hosts' detection algorithms are based on static thresholds, it was considered for the purpose of a simplified simulation process. However, fixed values of the utilisations are unsuitable for a dynamic Cloud environment which changes frequently and has an unpredictable workload. Therefore, it's important to develop advanced overloaded and underloaded hosts detection algorithms that are based on dynamic thresholds, which should be able to dynamically adjust according to the workload patterns expedited by the VMs.

Another potential future project is implementing the proposed framework in a real Cloud environment. Only few Cloud platforms offer VM live migration as a feature, and those commercial Cloud platforms have developed it for commercial proposes not for experiments. Most private Cloud platforms keep their technology hidden and prevent it from modifying. Therefore it not feasible to carry out largescale experiments on commercial Cloud platforms, and due to this reason, the proposed framework has been implemented in a CloudSim simulator.

OpenStack is an open source Cloud operating system which offers live VM migration as a feature, and implementing the proposed framework on OpenStack would be an interesting direction for future research.

According to the current functions of the proposed framework, it doesn't have a mechanism to understand the network topology of the data centre network. It migrates VMs to underloaded hosts based on host's memory and bandwidth, but doesn't have an understanding of the network topology. Due to this reason, migrating VMs may end up hosted on logically distant physical servers leading to costly data transfers. To eliminate this

problem, it is necessary to observe the topology of the network prior to VM migration in order to allocate the VMs on closely located hosts.

Scalability and elimination of a single point of failure are important benefits of designing an efficient and reliable system. The proposed framework adopted a centralised load balancing approach, and the Central Controller of the framework was deployed on a single host in the cluster. Due to that reason, this limits the scalability of the system and creates a single point of failure. To address this issue, it would be possible to run a second Central Controller in the cluster, but two Central Controllers need to be replicated regularly in order to provide an efficient and accurate service. A more complex research problem would be to synchronise activities of Central Controllers, and this problem is an interesting direction for future research, and its solution will provide a reliable system by eliminating a central point of failure.

References

1. Agrawal.N, Pateria.R.K, A Survey of Pre-Copy Live Migration Techniques in Virtual Network Environment, *International Journal of Engineering Research & Technology*, ISSN 2278-0181, Pages 2202-2207
2. Agrawal.N, Pateriya.R.K.(2013).Enhanced Time Series Based Pre-Copy Method for Live Migration of Virtual Machines. International Journal of Advances in Engineering & Technology .IJAET.ISSN:22311963.
3. Ahmadi.M.R, Maleki.D. (2010). Performance Evolution of Server Virtualisation in Datacentre Architecture. 2010 5th International Symposium Telecommunications (IST).Iran. IEEE Explorer. DOI: 10.1109/ISTEL.2010.5734102
4. Aiash.M,Mapp.G,Gemikonaki.G.(2014).Secure Live Machine Migration: Issues and Solutions.28th International Conference on Advanced Information Networking and Applications Workshop.IEEE. ISBN: 978-1-4799-2653-4
5. Akoush.S , Sohan.R , Rice.A , Andrew.W.M , Hooper.A . Predicting the Performances of Virtual Machine Migration. University of Cambridge .UK
6. Ameen.R.Y, Hamo.A.Y (2013).Survey of Server Virtualisation. International Journal of Computer Science and Information Security.Vol 11
7. Anala.M.R, Jyoti.S, Shobha.G (2013).A Framework for Secure Live Migration of Virtual Machine. Advanced in Computing, Communications and Informatics (ICACCI).ISBN 978-1-4673-6217-7
8. Antonopoulos.N, Gillam.L. Cloud Computing, Principles, Systems and applications (2010).Springer London.
9. Arista.com. The Impact of Virtualisation on Cloud Networking [online].Available at: https://www.arista.com/assets/data/pdf/VirtualClouds_v3.pdf [Accessed 18.09.2017]
10. Baghshahi.S.S,Jabbehdari.S,Adabi.S.(2014).Virtual Machines Migration based on Greedy Algorithm in Cloud Computing. International Journal of Computer Applications .Volume 96 .
11. Bagshani.S.S , Jabbehdari.S, Adabi.S, Virtual Machines Migration based on Greedy Algorithm in Cloud .International Journal of Computer Applications , 2014 , Volume 96 – N0v12, Pages 0975-8887
12. Banerjee.I, Moltmann, P,Tati.K,Venkatasubramanian.R, VMWare ESX Memory Resource Management:Swap [online] VMWare.com. Available at: http://libweb.anglia.ac.uk/referencing/harvard.htm [Accessed 10 February 2017]
13. Barbalace.A, Lu.P, Palmieri.R, Ravindran.B (2014). Adaptive Live Migration to Improve Load Balancing in Virtual Machine Enviornment.Virginia Tech.U.S.A .
14. Barry.D.K, Dick.D (2013) Web Services, Service Oriented Architectures, and Cloud Computing Second Edition. Morgan Kaufmann.USA
15. Beloglazov.A, Buyya.R, Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centres. Wiley InterScience,2011, DOI:10.1002/cpe.1867.
16. Beloglazov.A, Buyya.R.2011.Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual

Machines in Cloud Data Centers.Cloud Computing and Distributed Systems (CLOUDS)Laboratory Department of Computer Science and software Engineering, University of Melbourne, Australia .
17. Benmessaoud.N, Williams.C.J, Mudigonda.U.M, Tulloch.M (2014).Network Virtualisation and Cloud Computing. Microsoft Systems Center.
18. Benson.T,Akella.A, Maltz.D.A.(2010).Network Traffic Characteristics of Data Centres in the Wild. Proceeding of the 10th ACM SIGCOMM Conference on Internet Measurement.ACM.NY.ISBN:978-1-4503-0483-2.
19. Bhadani.A,Chaudhary.S (2010).Performance Evaluation of Web Servers Using Central Load Balancing Policy Over Virtual Machines on Cloud. Proceedings of the Third Annual ACM Bangalore Conference.Bangalore.India ACM:2010:1-4 16.
20. Blenk.A, Basta.A, Reisslein.M,Kellerer.W.(2015). Survey on Network Virtualisation Hypervisors for Software Defined Networking.IEEE Communications Surveys and Tutorials. vol. 18, no. 1, pp. 655-685, First Quarter 2016
21. Botero.D.P.(2012).A Brief Tutorial on Live Virtual Machine Migration From Security Prespective.USA.
22. Buyya.R, Broberg.J , Goscinski.G (2011) . Cloud Computing Principles and Paradigms. A Jhoon Willey & Sons. Canada
23. Calheiros.R.N, Rajan.R, Beloglazov.A, Rose.D.A.F.C, Buyya.R.(2011). CloudSim:a toolkit for modelling and simulation of cloud computing environments and evaluation of resource provisioning algorithms.Willey Online Library.
24. Chan.H, Chieu.T (2010) Ranking and Mapping of Applications to Cloud Computing Services by SVD.Network Operations and Management Symposium Workshops(NOMS Wksps 2010.Japan. Electronic ISBN: 978-1-4244-6039-7
25. Chen.Q, Chen.J, Zheng.B,Cui.J,Qian.Y.(2015).Utilisation-Based VM Consolidation Scheme for Power Efficiency in Cloud Data Centres. Communication Workshop (ICCW), 2015 IEEE International Conference On. London. IEEE.ISSN 2164-7038.
26. Chowdhury.N.M.M.K, Boutaba.R.(2009).A Survey of Network Virtualisation. Volume 54, Issue 5, 8 April 2010, Pages 862–876
27. cio.com .2012.Top Enterprise Applications for Public Clouds.[online] Available at: http://www.cio.com/article/2394798/developer/top-10-enterprise-applications-for-public-clouds.html.[Accessed 26 February 2015]
28. Cisco 2015.Cisco Global Cloud Index:Forecast and Methodology, 2014-2019 [online].Available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf [Accessed: 29.08.2016]
29. Cloud Adaptation Trends in UK Public Sector (2015).Outsourcery.co.uk (online).Available at : https://www.outsourcery.co.uk/media/1377/cif-outsourcery-white-paper-16.pdf [Accessed: 14.07.2017]
30. Cloud.google.com.2017. Google Compute Engine [online] Available at : https://cloud.google.com/compute/docs/instances/live-migration [Accessed 22.08.2017]
31. CloudSim simulation framework - Superwits academy (Online) Available at: http://www.superwits.com/library/cloudsim-simulation-framework [Accessed: 14 February 2017].

32. Configuration Maximum vSphere 6.5.VMware Inc [online].Available at: https://www.vmware.com/pdf/vsphere6/r65/vsphere-65-configuration-maximums.pdf [Accessed 22.08.2017]
33. Configuration Maximums VSphere 6.5, VMware.com.2017 (online). Available at: https://www.vmware.com/pdf/vsphere6/r65/vsphere-65-configuration-maximums.pdf [Accessed: 14.11.2017]
34. Corradi.A,Fanelli.M,Foschini.L.(2014).VM Consolidation: A Real Case Based on OpenStack Cloud. Future Generation Computer Systems.Volume 32.
35. Dargie.W .(2014) .Estimation of the Cost of VM Migration. The 23rd International Conference on . Computer Communication and Networks(ICCCN), 2014 .China . Electronic ISBN: 978-1-4799-3572-7.
36. Data Governance in the Cloud [online]. Cloudindustryforum.org. Available at: file:///C:/Users/k1419238/Downloads/CIF0029-Data-governance-in-the-Cloud%20(2).pdf [Accessed: 18.07.2017]
37. De Carvalho.C.A.B, De Castro.R.M.A, De Castro.M.F, Coutinho.E.F, Agoulmine.N (2017).State of the Art and Challenges of Security SLA for Cloud Computing. Computer & Electrical Engineering. Volume 59
38. Denneman.F, (Alternative) VM Swap File Location [online] Available at : http://frankdenneman.nl/2012/11/19/alternative-vm-swap-file-locations-qa-part-2/ [Accessed 10 February 2017]
39. Deshpande.U, Chan.D, Guh.D, Edourad.J, Gopalan.K, Bila.N, Agile Live Migration of Virtual Machines, IBM Research Centre, USA. 2016 IEEE International Parallel and Distributed Processing Symposium(IPDPS),2016
40. Dhanoa.I.S,Khurmi.S.S. (2015).Analysing Energy Consumption During VM Live Migration. International Conference on Computing Communication and Automation (ICCCA2015).India. IEEE ISBN 978-1-4799-8890-7 .
41. Dhingra.A, Paul.S, Green Cloud: Heuristic Based BFO Technique to Optimise Resource Allocation.Indian Journal of Science and Technology,2014, Vol7(5), Pages 685-691
42. Dhir.P, Garg.P (2017) . Survey on Data Masking Techniques. International Journal of Innovations & Advancement in Computer Science.Volume 4.ISSN 2347-8616
43. Dillon.T,Wu.C,Chang.E.(2010).Cloud Computing: Issues and Challenges. 24th IEEE International Conference on Advanced Information Networking and Applications. EEE. INSPEC 11399678
44. Dobson.S, Sterritt.R,Nixon.P (2010) Fulfilling the Vision of Autonomic Computing. Computer, the flagship publication of IEEE Computer Society. 2010. ISSN 0018-9106
45. Douglas.K.B (2017).Service Architecture [online].Service Architecture.Com.Available at: http://www.service-architecture.com/articles/web-services/web_services_explained.html [Accessed:01.07.2017]
46. Duggan.M, Flesk.K,Duggan.J,Howley.E,Barrett. A Reinforcement Learning Approach for Dynamic Selection of Virtual Machines in Cloud Data Centres. Innovative Computing Technology (INTECH),2016 Sixth International Conference. IEEE
47. Endrei.M, Ang.J, Arsanjani.A, Chua.S, Comte.P, Krogdahl, Luo.M, Newling.T .(2004) . Patterns: Service Oriented Architecture and Web Services.IBM Publishers.USA.

48. Eramo.V, Ammar.M, Lavacca.F.G (2017).Migration Energy Aware Reconfigurations of Virtual Network Function Instances in NFV Architectures. IEEE.ISSN 2169-3536.
49. Fan.X,Weber.W.D, Barroso.L.A, “Power provisioning for a warehouse-sized
50. Farrahi . M , Mohamed.C .2010 . Decreasing Live Virtual Machine Migration Down – Time Using a Memory Page Selection Based on Memory Change PDF
51. Garfinkel.S.(2011).The Cloud Imperative [online]. Available at: https://www.technologyreview.com/s/425623/the-cloud-imperative/ [Accessed: 13.07.2017] .
52. Giannakouris.K, Smihily.M (2016). Cloud Computing – Statistics on the Use by Enterprises. Eurostat Information [online]. Available at : http://ec.europa.eu/eurostat/statistics-explained/index.php/Cloud_computing_-_statistics_on_the_use_by_enterprises [Accessed: 14.07.2017]
53. Giannakouris.K,Shimly.M.(2016).Cloud Computing – Statistics on the use by enterprises[online].ec.europe.eu. Available at : http://ec.europa.eu/eurostat/statistics-explained/index.php/Cloud_computing_-_statistics_on_the_use_by_enterprises [Accessed: 10.08.2017]
54. Gill.A.Q (2015).Adaptive Cloud Enterprise Architecture. Intelligent Information Systems Volume 4. World Scientific Publishing Co. London.
55. Gomez.M.(2011). Simulation Tool for Market Based Cloud Resource Allocation. Birmingham University.UK
56. Greenberg.A, Hamilton.J, Maltz.D.A, Patel.P. 2009.The Cost of Cloud : Research Problems in Data Center Networks. Microsoft Research .pp 68-73 .
57. Greenberg.A, Hamilton.J, Maltz.D.A, Patel.P. 2009.The Cost of Cloud : Research Problems in Data Center Networks. Microsoft Research. pp 68-73 .
58. Haikun .L , Jin.H , Liao.X , Chen Yu , Cheng.Z . 2011 . Live Virtual Machine Migration via Asynchrobous Replication and State Synchronization . IEEE Transaction on Parallel and Distributed Systems .
59. Haikun.L, Jin.H, Liao.X, Hu.L Chen.Y, (2009).Live Migration of Virtual Machines Based on Full Sytem Trace and Reply. 18th ACM International Symposium on High Performance Distributed Computing.Germany.ISBN:978-1-60558-587-1
60. Hameed.A, Khoshkbarforoushha.A, Ranjan.R, Jayaraman.P.P, Kolodziej.J,Balaji.P, Zeadally.S. (2014).A Survey and Taxonomy on Energy Efficient Resources Allocation Techniques for Cloud Computing System.Springer-Verlag.Vienna.
61. Herman.B. (2016).Details of Anthem’s Massive Cyberattack remains in the Dark a Year Later [online].Modernhealthcare.com. Available at : http://www.modernhealthcare.com/article/20160330/NEWS/160339997 . [Accessed: 13.07.2017]
62. Hildred.T,Gordon.S,Jorm.D.(2012).Red Hat Enterprise Virtualisation 3.1.[online].rehat.com.Available at: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.1/html-single/Technical_Reference_Guide/index.html [Accessed: 15.08.2017]
63. Hines.R.M , Deshapande.U, Kartik.G , Post-Copy Live Migration of Virtual Machines. ACM SIGPOS Operating System Review, 2009, Volume 43 Issue 3 , Pages 14-26 .
64. Hu.R, Jingfei.J,Liu.G,Wang.L (2014). Efficient Resources Provisioning Based on Load Forecasting in Cloud. Scientific World Journal 2014.

65. Hu.W, Hicks.A,Zhang.L, Dow.E.M, Sony.V , Jiang.H, Bull.R, Jeanna.N, N.Matthews. A Quantity Study of Virtual Machine Live Migration. Clarkson University .New York.
66. Huang.K.C, Tsai.M.U, Lu.S.J, Hung.C.H (2016). SLA-Constrained Service Selection for Minimising Costs of Providing Composite Cloud Services Under Stochastic Runtime Performance. SpringerPlus. DOI 10.1186/s40064-016-1938-6
67. Huebcher.M.C, McCann.J.A.2006. A Survey of Autonomic Computing – Degrees, Models, and Applications. Imperial College London.
68. Igbe.D.(2013).How to Migrate an Instance with Zero Downtime: OpenStack Live Migration with KVM Hypervisor and NFS Shared Storage.[online].Mirantis.com. Available at: https://www.mirantis.com/blog/tutorial-openstack-live-migration-with-kvm-hypervisor-and-nfs-shared-storage/ [Accessed: 10.09.2017]
69. Information Economy Report 2013 – The Cloud Economy and Developing Countries.unctag.org. Available at : http://unctad.org/en/PublicationsLibrary/ier2013_en.pdf [Accessed : 10.09.2018]
70. Information Economy Report 2013 on Cloud Economy and Developing Countries [online].UNCTAD.org. Available at : http://unctad.org/en/PublicationsLibrary/ier2013_en.pdf . [Accessed: 24.07.2017]
71. Ismail.N (2017).UK Cloud Adaptation Rate Reaches 88% - Cloud Industry Forum [online].Information-age.com. Available at : http://www.information-age.com/uk-cloud-adoption-reaches-88-123464995/ [Accessed: 14.07.2017]
72. Jasma.K (2013). Cloud Computing. Jones & Bartlett. USA
73. Josuttis.N.M (2007). SOA in Practice: The Art of Distributed System Design (Theory in Practice), 2007, O'reilly.
74. Joysula.V , Orr.M , Page.G (2012) Cloud Computing : Automating the Virtualised Data Center. Cisco Press. U.S.A , p 263-p270)
75. Kalin.M (2013).Java Web Services: Up and Runiing, 2nd Edition.O'Relly Media Inc.ISBN 9781449373856.
76. Kapil.D, Emmanuel.S,Pilli, Joshi.J.C (2013) .Live Virtual Machine Migration Techniques: Survey and Research. 3rd IEEE International Advance Computing Conference(IACC).
77. Kavis.M (2016). Google Provides a Glimpse Into a The Future of Cloud Computing.Forbes.com (online).Available at: https://www.forbes.com/sites/mikekavis/2016/03/25/google-provides-a-glimpse-into-the-future-of-cloud-computing/#185b1e917056 [Accessed : 23.07.2017]
78. Kay.R(2008).Storage Virtualisation [online].Available at http://www.computerworld.com/article/2551439/data-center/storage-virtualization.html [Accessed : 21.06.2017]
79. Kejiang.Y, Xiaohong.J,Ma.R,Yan.F.(2012).VC-Migration: Live Migration of Virtual Clusters in the Cloud.13th International Conference on Grid Computing. China. ACM/IEEE. ISBN: 978-1-4673-2901-9.
80. Kerner.S.M.(2016).92 Percent of Data Centre Traffic will be Cloud by 2020 [online]. Available at: http://www.enterprisenetworkingplanet.com/datacenter/92-percent-of-data-center-traffic-will-be-cloud-by-2020.html [Accessed 04.04.2018] .
81. Kolyshkin, K. "Virtualization in Linux." 2006 .Available at http://download.openvz.org/doc/openvz-intro.pdf.[Accessed : 10.11.2016]

82. Konrad.A (2015). Report: Cloud Market Cap To Pass $500 Billion by 2020.Forbes.Com (online).Available at: https://www.forbes.com/sites/alexkonrad/2015/06/18/byron-deeter-state-of-the-cloud/#2be46dbe767c. [Accessed:26.07.2017]
83. Kouki.Y, Lina.E.I , (2012). SLA Driven Capacity Planning for Cloud Applications.*IEEE 12th International Conference on Cloud Computing Technology and Science*. P 135
84. Kouki.Y, Lina.E.I , (2012). SLA Driven Capacity Planning for Cloud Applications.IEEE 12th International Conference on Cloud Computing Technology and Science. P 135
85. Kumar.R, Charu.S (2015). An Importance of Using Virtualising in Cloud Computing. Global Journal of Computers and Technology.ISSN 2394501X
86. Leelipushpam.P.G.J,Sharmila.J.(2013).Live VM Migration Techniques in Cloud Enviornment – A Survey. 2013 IEEE Conference on Information & Communication Technologies. India. Electronic ISBN: 978-1-4673-5758-6
87. Liu, H., Jin, H., Liao, X., Hu, L., Yu, C. (2009).Live migration of virtual machine based on full system trace and replay. In: Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing, pp. 101–110
88. Liyanage.S,Khaddaj.S,Francik.J.(2015).Virtual Machine Migration Strategy in Cloud Computing.*14th International Symposium on Distributed Computing and Applications for Business Engineering and Science.* IEEE.China.ISBN:978-1-4673-6593-2
89. Lizhe Wang, Jie Tao, Gregor von Laszewski, Holger Marten. *Multicores in Cloud Computing: Research Challenges for Applications,* Journal of Computers, Vol
90. Llorente.I.M, Montero.M.S, Huedo.E, Leal.K. 2006. A Grid Infrastructure for Utility Computing. *Infrastructure for Collaborative Enterprises, 2006. WETICE '06'. 15th IEEE International Workshops.Manchester.pp 163-168.*
91. Lowe.S.(2011).Block Level Storage vs File Level Storage: A Comparison [Online] .Techrepublic.com. Available at: https://www.techrepublic.com/blog/the-enterprise-cloud/block-level-storage-vs-file-level-storage-a-comparison/ [Accessed: 08.06.2015]
92. Ma.J, (2015).Top 10 Security Concerns for Cloud-Based Services [online].Available at: https://www.incapsula.com/blog/top-10-cloud-security-concerns.html [Accessed: 17.07.2017]
93. Magoules.F, Pan.J, An Tan.K, Kumar.A (2009) Introduction to Grid Computing.Chapmon & Hall Publishers.ISBN 13-978-1-4200-7407-9
94. Mandal.U, Habib.M.F, Zhang.S, Tornatore.M, Mukherjee.B (2013). Bandwidth and Routing Assignment for Virtual Machine Migration in Photonic Cloud Networks. *Optical Communication (ECOC 2013), 39th European Conference and Exhibition. UK.ISBN:978-1-84919-759-5*
95. Mashtizadeh.A, Celebi.E, Garfinkel.T, Cai.M, (2013). The Design and Evolution of Live Storage Migration in VMware ESX. VMware Inc.
96. Miguel.L, Lorenzo.B, Yannis.A, Dimitriadis, Eduardo.G.S.2004.Grid Characteristics and Uses: A Grid Definition. School of Telecommunications Engineering, University of Valladoid.Spain

97. Mishra.N.K, Mishra.N (2015). Load Balancing Techniques: Need Objectives and Major Challenges in Cloud Computing - A Systematic Review. *International Journal of Computer Applications (0975 – 8887)* Volume 131 – No.18, December 2015.

98. Mochizuki.K,Yamazaki.H,Misawa.A.(2013).Bandwidth Guaranteed Method to Relocate Virtual Machines for Edge Cloud Architecture.*15th Asia-Pacific Network Operations and Management Symposium(APNOMS).*IEEE.Japan.ISBN:978-4-8855-2279-6.
99. Moghaddam.F.F,Cheriet.M,Nguyen.K.K .(2011).Low Carbon Virtual Private Clouds. 4th International Conference on Cloud Computing .IEEE.USA.
100. Mulesoft.com. Evolution of SOA [online].Available at:file:///C:/Users/k1419238/Downloads/soa-whitepaper-updated.pdf [Accessed 12.07.2018]

101. Murugesan.S, Bojanova.I (2016). Encyclopaedia of Cloud Computing.Jhon Wiley & Sons.USA
102. Nathan.S , Purushottam.K , Bellur.U , 2013 . Resource Availability Based Performance Benchmarking of Virtual Machine Migration .Bombay. India.
103. National Institute of Standards and Technology.2011.The NIST of Cloud Computing [online] Available at: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. [Accessed 19 June 2014]
104. Palit.H.N, Li.X, Lu.S, Larsen.L.C, Setia.J.A (2013) Evaluating Hardware Assisted Virtualisation for Deploying HPC as a Service.*7th International Workshop on Virtualisation Trchnologies in Distributed Computing*. New York .ISBN 978-1-4503-1985-0
105. Patel.P.D ,Karamat.M , Bhavsar.M .D , Potdar.M.B .2009.Live Virtual Migration Techniques in Cloud Computing: A Survey .*International Journal of Computer Applications* .Beijin. China.
106. Peterson.L,Davie.B , (2011).Computer Networks 5th Edition .USA.
107. Pothuri.A, Shaik.S. (2016). Ensuring an Efficient and Reliable Quality of Service in Cloud Computing. *International Journal of Advance Research in Computer Science and Management Studies*.ISSN:2321-7782.
108. Prakash.S, Prakash.D (2014).A Hybrid GABFO Scheduling for Optimal Makespan in Computational Grid. *International Journal of Applied Evolutionary Computation (IJAEC).*
109. Ragan.S (2014). Code Spaces Forced to Close its Doors After Security Incident. Salted-Hash-Top Security News [online].Available at: http://www.csoonline.com/article/2365062/disaster-recovery/code-spaces-forced-to-close-its-doors-after-security-incident.html [Accessed: 18.07.2017]
110. Raj.P. (2012). Cloud Enterprise Architecture. Taylor & Francis Group. USA
111. Rajyashree, Richhariya.V. (2015).Double Threshold Based Load Balancing Approach by Using VM Migration for the Cloud Computing Environment. *International journal of Engineering and Computer Science ISSN: 2319-7242*
112. Rashidi.B, Sharifi.M, Jafari.M. (2013). A Survey on Interoperability in the Cloud Computing Enviornment. *I.J Modern Education and Computer Science*, p 17-13
113. Redhat.com.2017.What is OpenStack? [Online]. Available at: https://www.redhat.com/en/topics/openstack [Accessed 23.08.2017]

114. Rimal.B.P, Choi.E, Lumb.I (2010).A Taxonomy, Survey, and Issues of Cloud Computing Ecosystem.Springer.
115. Rose.R (2004).Survey of System Virtualisation Techniques.School of Electrical Engineering and Computer Science.
116. Sabharwal.N,Wali.P (2013).Cloud Capacity Management .Springer. U.S.A
117. Sahoo.J, Mohapatra.S, Lath.R. Virtualisation: A Survey on Concepts, Taxonomy and Associated Security Issues. Computer and Network Technology (ICCNT), 2010 International Conference.IEEE. ISBN 978-1-4244-6962-8
118. Scarfone.K, Souppaya.M, Hoffman.P (2011). Guide to Security for Full Virtualisation Technologies. NIST Special Publication.USA.
119. Schwiegelshon.U, Badia.R.M,Bubak.M,Danelutto, Dustdar.S .2010. Perspectives on Grid Computing .Future Generations Computer Systems.pp 1104-115
120. Sefraoui.O, Aissaoui.M,Eleuldj. OpenStack: Toward an Open-Source Solution for Cloud Computing.2012.Volume 55. Pp 38-42
121. Shaine.G. (2014). 100 Gbps Headed for the Data Centre [online]. Available at: http://www.networkcomputing.com/data-centers/100-gbps-headed-data-center/407619707 [Accessed 15.10.2017]
122. Shawish.A, Salama.M.(2014)Cloud Computing :Paradigms and Technologies. Inter-cooperative Collective Intelligence. Germany. ISBN 978-3-642-35015-3
123. Shields, b. (2011) Virtual memory management techniques: A beginner's guide.[online] Available at: http://searchservervirtualization.techtarget.com/tip/Virtual-memory-management-techniques-A-beginners-guide [Accessed: 10 February 2017].
124. Simpson.E.(2017) Evaluation of Cloud Computing Services Based on NIST 800-145 [online].Available at : https://www.nist.gov/sites/default/files/documents/2017/05/31/evaluation_of_cloud_computing_services_based_on_nist_800-145_20170427clean.pdf [Accessed: 20.5.2016]
125. SINGH, R. (2015) Delays in Transmission. (Online)Available at: http://com2networks.blogspot.co.uk/2013/09/delays-in-transmission.html [Accessed: 13 February 2017].
126. SinghA,Korupolu M, MohapatraD. Server-storage virtualization: Integration and Load Balancing in Data Centres.Proceedings of the 2008 ACM/IEEE Conference on Supercomputing.USA.IEEE.ISBN. 978-1-4244-2834-2
127. Sinti.J , Jiffry.F , Aiash.M . 2014 . Investigating the impact of Live Migration on the Network Infrastructure in Enterprise Environment. 28th International Conference on Advanced Information Networking and Applications Workshop ISBN 978-1-4799-2652-7 p 154-156
128. Song.X, Ma.Y, Teng.D (2015) . A Load Balancing Scheme Using Federate Migration Based on Virtual Machines for Cloud Simulations. Mathematical Problems in Engineering.Article ID505432
129. S. Srikantaiah, A. Kansal, and F. Zhao. Energy Aware Consolidation for Cloud Computing. In Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08, pages 10–10, Berkeley, CA, USA, 2008. USENIX Association.

130. STAGE.A , SETZER.T, Network-Aware Migration Control and Scheduling of Differentiated Virtual Machine Workloads. Cloud 09 Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges in Cloud Computing, 2009, ISBN 978-1-4244-3713-9, 9-14.
131. Stahl.E , Corona.A , De Gillio.D , Demuro.M , Dowling.A , Duijvestin.L, Fernandes.A, Jewell.D, Keshavamurthy.B, Markovits.S, Mouleeswaran.C, Raess.S, Yu.K (2013) Performance and Capacity Themes for Cloud Computing . IBM . U.S.A p35-52.
132. Storage Virtualisation in a Storage Area Network.Google.com (online).Available at: https://www.google.com/patents/US6898670 [Accessed 01.07.2015]
133. Suetake.M,Kizu.H, Kourai.2016.Split Migration of Large Memory Virtual Machines . University of New Souhtwales, Australia.
134. Svard.P, Walsh.S, Hunzia.B, Tordsson.Y, Elmroth.E (2014).The Noble Art of Live VM Migration- Principles and Performance of Pre-Copy, Post-Copy and Hybrid Migration of Demanding Workloads.
135. Systems (HotPower), 2008, pp. 1–5.
136. Tate.J, Beck.P, Ibarra.H.H, Kumaravel.S, Miklas.L (2016).Introduction to Storage Area Networks.IBM Publishers.USA.
137. Thiruvenkadam.T.Kamalakkannan.P (2015).Energy Efficient Multi-Dimensional Host Load Aware Algorithm for Virtual Machine Placement and Optimization in Cloud Environment. Indian J Sci Technol.2015-8(17) 1-11
138. Tian W, Xu M, Chen Y, Zhao Y. (2015).Prepartition: A new paradigm for the Load Balance of Virtual Machine Allocation in Data Centres. Computer Science Magazine.
139. Tian.W,Zhao.Y,Zhong.Y,Xu.M,Jing.C.(2011).A Dynamic Integrated Load-Balancing Scheduling Algorithm for Cloud Data Centres. International Conference on Cloud Computing and Intelligence System.IEEE.ISBN:978-1-61284-204-2
140. UK Cloud Adaptation & Trends for 2016. Cloudindustryforum.org [online].Available at: https://www.cloudindustryforum.org/content/uk-cloud-adoption-trends-2016 [Accessed: 01.09.2017]
141. Understanding Full Virtualisation, Paravirtualisation, and Hardware. [online].VMware.com. Available at: https://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html. [Accessed 15.06.2017]
142. Understanding Processor (% Processor Time)and Process (%Processor Time)-Microsoft.com (online) Available at: https://social.technet.microsoft.com/wiki/contents/articles/12984.understanding-processor-processor-time-and-process-processor-time.aspx [Accessed: 10 February 2017]
143. Varian.M (1997).VM and the VM Community:Past, Present, Future. Share Inc.
144. Venkatesh.S, Shatrugna.R , Kintali.S .2009. Survey of Virtual Machine Migration Techniques. Department of Computer Science , U.S.A .
145. Vernon.I, Folorunso.F . 2012. Cloud Computing. [Online]. [1 June 2017]. Available at : https://docs.google.com/presentation/d/1cTxlqYethME-M4gvV5aNLhPVWToPmabAPc7ixCwFIZU/edit#slide=id.p [Accessed: 29.08.2016]
146. virtualizationreview.com.2009.Type 1 and Type 2 Hypervisors Explained.[online] Available at: http://virtualizationreview.com/blogs/everyday-

virtualization/2009/06/type-1-and-type-2-hypervisors-explained.aspx [Accessed 27.02.2015]
147. Vmware.com.2017.vSphere with Operation Management [online] Available at: https://www.vmware.com/uk/products/vsphere/vmotion.html [Accessed 23.08.2017]
148. Voorsluys.W , Brobeg.J , Venugopal.S , Buyya .R . 2009 . Cost of Virtual Machine Live Migration in Clouds: Performance Evaluation. University of Melbourne, Australia.
149. Vyas.J, Modi.p. (2017).Providing Confidentiality and Integrity on Data Stored in Cloud Storage by Hash and Meta-Data Approach. International Journal of Advance Research in Engineering, Science & Technology. Volume 4.Issue 5.P-ISSN 2394-2444
150. Waltenegus.D, (2014).Estimation of the Cost of VM Migration.23rd International Conference on Computer Communication Networks (ICCCN).China. ISBN:978-1-4799-3572-7
151. WeiTek.T, XiaoYing.B, Yu.H (2014).Software as a Service (SaaS): Perspectives and Challenges. Science China Information Sciences. Volume 57, Issue 5, pp1-5
152. Wood.T,Shenoy.P,Venkataramani.A.(2009).Sandpiper:Black Box and Gray Box Resource Management for Virtual Machines. Computer Networks.Volume 53.Pages 2923-2938
153. Xu.M, Tian.W,Buyya.R.(2010).A Survey on Load Balancing Algorithms for Virtual Machines Placement in Cloud. Wiley Inter Science.DOI:10.1002/cpe.
154. Y. Wang, Y. Cui, P. Tao, H. Fan, Y. Chen, and Y. Shi, “Reducing shared cache contention by scheduling order adjustment on commodity multicores,” in Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing, 2011.
155. Yang K, Gu J, Zhao T, SunG. An optimized control strategy for load balancing
156. Zafar.F, Khan.A, Malik.S.R, Ahmed.M, Anjum.A, Khan.M.I, Javed.N, Alam.M, Jami.F (2017). A Survey of Cloud Computing Data Integrity Schemes: Design Challenges, Taxonomy and Future Trends. Computers and Security. Volume 65, pp 29-49.
157. Zhang.H, Yang.X.(2012).Cloud Computing Based on SOA.12tth International Symposium on Computational Intelligence and Design.2012. IEEE.China.ISBN978-1-4673-2646-9
158. Zhang.J , Ren.F , Chuang.L .2014 . Delay Guaranteed Live Migration of Virtual Machines .IEEE Conference on Computer Communications.
159. Zhang.K. 2013.Openstack VM Live Migration [online]. Available at: https://kimizhang.wordpress.com/2013/08/26/openstack-vm-live-migration/#LiveMigration-Blocklivemigration [Accessed 24.08.2017]
160. Zhang.Q, Cheng.L, Boutaba.R (2010). Cloud Computing: State of the art and research challenges. Journal of Internet Services and Applications. Volume 1.Issue 1.pp 7-18
161. Zhao.Y, Huang.W.(2009).Adaptive Distributed Load Balancing Algorithm Based on Live Migration of Virtual Machines in Cloud . Fifth International Joint Conference on INC, IMC and IDC.Korea.IEEE.ISBN978-1-4244-5209-5.