Optimized Security Aware VM Placement Algorithms

by

Motlatsi Isaac Thulo

Submitted in partial fulfillment of the requirements for the Masters Degree
(Computer Science)
in the Faculty of Engineering, Built Environment and Information Technology
University of Pretoria, Pretoria

October 2019

Publication data:

Motlatsi Isaac Thulo. Optimized Security Aware VM Placement Algorithms. Masters Degree dissertation, University of Pretoria, Department of Computer Science, Pretoria, South Africa, October 2019.

Optimized Security-aware VM Placement Algorithms

by

Motlatsi Isaac Thulo
E-mail: u15277489@tuks.co.za

Abstract

The rapidly increasing dependency on use of clouds results in Cloud Service Providers (CSPs) having to deal with high cloud services demands. To meet these demands, CSPs take advantage of virtualization technology to provide a seemingly unlimited pool of computing resources. This technology consolidates multiple instances of Virtual Machines (VMs) into the same Physical Machines (PMs) and share physical computing resources. To guarantee customer satisfaction, CSPs need to ensure optimized cloud environment that provides good Quality of Services (QoS) which conform to the performance levels stipulated in Service Level Agreements (SLAs). However, vulnerabilities associated with virtualization make it difficult to ensure optimization, more especially in multi-tenant clouds. In multi-tenant clouds, there are possibilities of consolidating VMs belonging to adversary users into the same PMs. This promotes inter-VM attacks that take advantage of shared resources to either spy, disrupt or corrupt co-located VMs. With this regard, it is important to consider placement of VMs in a manner that minimizes inter-VM attacks. This placement must, however, ensure initial objectives of providing good QoS.

The aim of this study is to implement a VM placement algorithm that reduces architectural vulnerabilities brought by multi-tenancy while observing optimization objectives. It focuses on currently available VM placement algorithms and evaluates them to identify the algorithm that assumes highest optimization objectives. The identified VM placement algorithm is further augmented with security features to implement Optimized Security-aware (O-Sec) VM Placement algorithms. CloudSim Plus is used to evaluate and validate the implemented O-Sec VM placement algorithms. The evaluations in this study show that O-sec VM placement algorithm retains optimization objectives inherited from the identified VM placement algorithm. This is an algorithm that is augmented towards O-sec VM placement algorithm.

Keywords: Cloud Computing Security, Optimized VM Placement Algorithms, Security-aware VM Placement Algorithms.

Supervisors : Prof. J. H. P. Eloff

Department : Department of Computer Science
Degree : Bachelor of Science (Honours)

"In a virtualized environment, the security level of the entire system is the same level as the security of the weakest virtual machine."

Mazhar Ali, Samee U Khan and Athanasios V Vasilakos (2015)

Acknowledgements

I would like to thank the following people for contributing towards the completion of this research:

- Professor J.H.P. Eloff for giving me an opportunity to work as his students on this research. I am grateful for his ceaseless guidance, motivation and supervision throughout this study. He worked as my mentor in both my studies and work experience while at University of Pretoria.

- 'Mamakhamise Thulo (Spouse) who provided endless support and motivation while studying towards completion of this research. She made me believe everything is possible even when situations could not allow.

- Moseme Thulo and Manapo Namane who provided timely proof-reading of this dissertation.

- Lastly, I would like to express my sincere gratefulness to my entire family for their continuous encouragement.

Contents

List of Figures

List of Graphs

List of Algorithms

List of Tables

Chapter 1

Introduction

Cloud computing is a specialized distributed computing paradigm that has seemingly unlimited pool of computing resources. It provides on-demand computing services such as pay-as-you-use storage, online applications and processing activities [9]. The cloud computing services are accessed through the internet using tools such as browsers and Application Programming Interfaces (APIs) [4]. These services gained popularity over the past recent years due to their convenience and ease of use. This popularity is influenced by several factors which include, among others, reduced costs of hiring readily available computing resources [28]. In addition, the little effort required for cloud end-users to set-up and get ready to use cloud resources also contribute to the popularity of this paradigm. Due to these factors, there are rapid migrations of computing activities from traditional use of desktops and server rooms into the cloud. In the cloud, 'secure' locations are provided to host applications and synchronize data generated by inter-connected electronic devices such as cellphones, tablets and laptops [17].

In recent years, there has been a rapid increase in the dependency on cloud computing services by both individuals and corporate world. For example, individuals make use of cloud applications to store their personal information such as videos, photos, and other important documents [17]. These applications include, among others, YouTube, Flickr and Dropbox. Similarly, organizations migrate their critical applications from their datacentres into the cloud. In addition, these organizations also make use of online applications such as Google Apps and Microsoft Office 365 that are provided by third party Cloud Service Providers (CSPs) [1]. In all these cases, CSPs need to ensure good Quality of Services (QoS) to allow acceptable application performances. This includes access to applications and other cloud resources upon request and with acceptable response times. That is, access to these applications and cloud resources need to meet expected minimum performance levels stipulated in the Service Level Agreement (SLA) [4].

The rapid growth and dependency on the use of cloud services has resulted in the need for bigger cloud datacentres. As mentioned, cloud applications and inter-connected electronic devices generate massive amounts of data that need to be stored in the cloud [17]. Due to the massively increasing amounts of data, CSPs incur high costs of buying expensive infrastructure in order to provide seemingly unlimited pool of computing resources. These infrastructure include, among others, high capacity servers, storage devices and high capacity network devices [28]. They also experience additional costs of administration and maintenance of these infrastructure. In order to sufficiently maintain provisioning of cloud services, these CSPs need to find ways to reduce costs of running cloud datacentres.

CSPs make use of technology called virtualization to successfully provide a seemingly unlimited pool of computing resources at lower costs. Through virtualization, multiple instances of operating systems run within the same Physical Machine (PM) [29]. They share computing resources such as CPU, memory and storage. To reduce costs, for example, virtualization consolidates Virtual Machines (VMs) into fewer number of PMs and switch off idle PMs [48]. This minimizes energy consumption within cloud datacentres and therefore reduces running costs. Virtualization also allows full utilization of computing resources within cloud infrastructure [49]. Consequently, there is a higher return on investment in the use of resources as compared to traditional computing where one operating system and/or application resides within one PM. However, some challenges are caused by the consolidation of VMs into the same PMs. These challenges include possibilities of over-utilization of computing resources due to variations of workloads within VMs [48]. This normally results in performance degradations of cloud services. In order to maintain good QoS, virtualization allows dynamic assignment and reassignment of VMs based on their resource requirements [11]. In this case, VMs are migrated from over-utilized PMs into those with enough resources to host them.

Virtualization uses software commonly known as a hypervisor to successfully host multiple VMs within the same PMs. This software provides an abstraction of the underlying hardware to mimic the actual PM environment [32]. It uses ‘containers’ to host multiple instances of operating systems. In principle, hypervisors provide full isolation between VMs residing within the same PM. However, this isolation is only logical and therefore is easily broken in practice [25]. This breakage is caused by number of known vulnerabilities such as undesired co-location of VMs.

In summary, CSPs experience high costs of construction, maintenance and administration of large cloud computing datacentres. In order to sustain provision of the seemingly infinite pool of computing resources, CSPs need to optimize the cloud environment. That is, to reduce costs of running datacentres while providing good QoS that satisfies paying customers. To achieve these, CSPs make use of virtualization as one of their measures. They consolidate VMs into less PMs to reduce energy consumption costs. This consolidation, however, compromises the security of VMs within the cloud infrastructure. It normally results in undesired co-locations that lead to security risks. There are on-going research efforts that attempt to minimize these security risks. Among them are those that securely place VMs within cloud infrastructure to avoid undesired co-locations. These research efforts modify hypervisor schedulers (VM placement algorithms) to allow placement of VMs that suit particular security objectives. These objectives include isolation of VMs belonging to adversary users [2], and separation of VMs based on their security levels [76].

Although security objectives reduce risks brought by virtualization, they also introduce other cloud specific challenges. These cloud specific challenges are those that jeopardize optimization objectives observed in cloud computing, i.e. the increased costs of running cloud datacentres [20]. In general, there is normally a trade-off between optimization and secure placement of VMs within the cloud. That is, the highly optimized cloud normally has limited security objectives and vice-versa. In order to bridge the gap, this research endeavors towards the notion of optimized security-aware VM placement algorithm. This is a VM placement algorithm that considers both security and optimization objectives in its placement strategy. In order to minimize the security risks, this VM placement algorithm firstly minimizes undesired co-location of VMs within the PMs. It achieves this by placing VMs of different vulnerability levels into different PMs. Secondly, the algorithm separates VMs belonging to adversary users to avoid possible inter-VM attacks. The optimization objectives observed are traffic cost minimization and energy consumption minimization.

1.1 Thesis Statement

Based on the introduction discussed in this chapter, the main claim of this research is formulated as follows:

Both security and optimization objectives can be observed in implementation of VM placement algorithms.

1.2 Problem Statement

Initially, VM placement algorithms focused solely on optimization of cloud services. They observed optimization objectives such as; power consumption, load balancing, response time, resource utilization, and SLA violations. These took advantage of virtualization to consolidate multiple VMs into fewer number of PMs. However, this consolidation of VMs introduced cloud specific vulnerabilities. These vulnerabilities include, among others, co-location of VMs belonging to adversary users. Failure to counter for these vulnerabilities results in compromise on confidentiality, integrity and availability of VMs. In order to minimize these vulnerabilities, some research efforts introduce security aspects in placement of VMs. These security-aware VM placement algorithms strive to reduce risks brought by cloud specific vulnerabilities. However, most of the currently available security-aware VM placement algorithms focus solely on security. They do not consider optimization objectives that were initially observed. Based on all these, it is concluded that there are two types of VM placement algorithms. These are VM placement algorithms that focus on optimization and those that focus on security. So, is it possible to implement a security-aware VM placement algorithm that takes into consideration optimization objectives that guarantee good QoS at reduced costs and without violating

SLA?

The aim of this study is to implement an optimized security-aware VM placement algorithm. This algorithm will take into consideration both security and optimization objectives. It will ensure secure placement of VMs to minimize some of the currently available security risks available in cloud infrastructures. In addition, it will also ensure good QoS at reduced costs and without violating the SLA.

1.3 Research Questions

The problem statement is addressed by answering this main research question:
How can an optimized security-aware VM placement algorithm be designed, developed and implemented without jeopardizing optimization objectives that ensure good quality of services, at reduced costs and without violating SLA?

To successfully achieve the aim of this study, this research needs to answer fundamental questions that arise as sub-problems:

- *How is cost reduction and QoS achieved in implementation of VM placement algorithms?*

This requires thorough literature review of currently implemented VM placement algorithms. It further requires identification and in-depth study of VM placement algorithms that strive towards providing good QoS and/or minimizing costs. This include studies on challenges associated with cost reduction and QoS objectives.

- *How can cost reduction and QoS be integrated into security-aware VM placement algorithms?*

In order to successfully implement VM placement algorithm that include cost reduction, QoS and security, a particular approach needs to be identified.

- *What are security requirements for optimized security-aware VM placement algorithm?*

This also requires literature review of currently available security risks in cloud computing. It further requires identification of vulnerabilities that introduce these risks. These will help identify countermeasures that can be included in placement of VMs to reduce these risks.

- *What are hypervisors and how are they involved in placement of VMs within the cloud?*

Thorough literature review of hypervisors is required. This includes studies on vulnerabilities brought by hypervisors. Also, how they contribute to placement of VMs within the cloud infrastructure.

- *How are hypervisors changed to allow VM placement that includes cost reduction, QoS and security?*

This requires identification of hypervisor components that contribute to placement of VMs within the cloud infrastructure. Furthermore, it requires replacement of these components, if possible, in order to change the default placement strategy.

1.4 Research Methodology

The following are steps taken to address the problem stated in Section 1.2.

In the *first step*, thorough literature review of currently implemented VM placement algorithms is conducted. To achieve this, VM placement algorithms are searched on known databases and websites starting from those published as early as 2009. This is due to the fact that most of the currently implemented VM placement algorithms are published within the last decade. The objectives reflected in implementation of VM placement algorithms are reviewed to enable classification into different groups.

The *second step* is assessment of the currently implemented VM placement algorithms. In this step, limitations of VM placement algorithms are identified and critically assessed. Furthermore, the currently implemented VM placement algorithms are evaluated based on objectives perceived in their implementation. This step helps in requirement elicitation, which builds towards design of the VM placement algorithm to be implemented in this research.

The *third step* is design phase of an optimized security-aware VM placement algorithm. In this phase, gathered requirements are used to create activity diagrams and pseudo-code for the algorithm to be implemented. The design phase is influenced by results obtained in evaluation of the currently implemented VM placement algorithms. This is because the VM placement algorithm that assumes highest required objectives is used as base in the design and implementation of optimized security-aware VM placement algorithm. It is selected to further be augmented towards the goal of this study.

The *fourth step* is the design, development and implementation of optimized security-aware VM placement algorithm. This step is incremental, with one identified objective added to the implementation of this algorithm at a time. Experiments are conducted in each stage of this implementation. The idea is to observe the impact of each objective to the overall VM placement algorithm behaviour.

The *fifth and the final step* is the validation of this optimized security-aware VM placement algorithm. This involves evaluation of the implemented VM placement algorithm based on physical resources usage within the cloud architecture. This happens iteratively throughout the implementation phase of the algorithm. CloudSim Plus is used in this step as a tool to evaluate optimized security-aware VM placement algorithm.

1.5 Scope and context of the study

The research at hand is limited to placement of VMs within the cloud architecture. Therefore, this study focuses solely on secure placement of VMs within the cloud architecture. It does not include other possible security measures that can be implemented to provide a more secure cloud computing environment.

1.6 Terminology used in the dissertation

This section provides brief definitions of the keywords or terms that are used in this dissertation:

Cloud Computing: new computing paradigm that provides on-demand and seemingly limitless pool of resources upon request, which requires little effort for end-user to set-up and get ready to use [28]. The computing resources are distributed over the network and normally accessed through use of the internet.

Cyber-security: is explained as protection of computer related assets [50]. These assets include computer systems, their hardware infrastructure and all supporting software. In the context of this study, cyber-security aims to protect computer assets against possible disruptions, corruption or spying that are brought by co-locations of instances in cloud computing.

Virtualization: technique that allows multiple instances of operating systems, known as VMs, to run within the same PM and share physical resources such as memory, CPU and bandwidth [35]. The number of VMs to be hosted in each PM depends on total resource requirements of VMs and resource capacities of the PM itself.

VM placement algorithms: algorithms that provide strategic assignment of each VM to PMs based on availability of physical resources [76]. This strategic placement depends on, in addition to the availability of physical resources, the objectives reflected on the algorithms (e.g. minimum energy consumption, machine count, CPU consumption, security, etc).

Optimization: the term in this study refers to strategic placement of VMs within the cloud infrastructure in order to achieve desired objectives. These objectives strive towards providing good QoS to paying customers at reduced costs. They include, among others: energy consumption minimization, load balancing, traffic cost minimization, and SLA violations minimization.

Security-aware VM placement algorithms: strategic placement of VMs that observes as its main objective, minimization of risks that are brought by clouds' architectural vulnerabilities.

1.7 Thesis Outline

The thesis consists of 10 chapters and are as follows:

- **Chapter** 1 is the introduction to the thesis.

- **Chapter 2** covers background information on cloud computing. It provides detailed literature on what cloud computing is and how its infrastructure is structured.

- **Chapter 3** covers literature on cloud computing security. It provides detailed information on the need for cloud computing security.

- **Chapter 4** covers literature on VM placement algorithms. It provides detailed background on the origin of the algorithms. It further shows their different categories to be used in this research.

- **Chapter 5** discusses different VM placement algorithms that are selected to further be evaluated in this research. The aim is to find the best that will further be augmented towards the objective of this study.

- **Chapter 6** covers the evaluation process that is used to find the best VM placement algorithm that qualifies to further be augmented.

- **Chapter 7** discusses in detail the selected VM placement algorithm (TPVMP) that qualifies to be augmented. It further shows how this algorithm is augmented to implement O-Sec VM placement algorithm.

- **Chapter 8** covers testing environment that is used to evaluate the implemented O-Sec VM placement algorithm.

- **Chapter 9** covers the evaluation and validation process of the implemented VM placement algorithm.

- **Chapter 10** gives conclusion of this research based on the results obtained from evaluation and validation.

Chapter 2

Cloud Computing

The theme of this research is cyber-security within the cloud. It concentrates on secure placement of VMs within the cloud to minimize cloud specific architectural vulnerabilities. In order to understand this, there is a need for detailed discussions on background information related to cloud computing. This includes discussions on what cloud computing is, where it originates and how it is architecturally structured.

2.1 What is Cloud Computing?

In the past decade, there has been a significant change in the world of computing [9]. Some computing services, such as applications and storage, migrate from local desktops and corporate server units into the 'cloud'. It is common practice that organizations hire on-demand pay-as-you-use computing devices, software or platforms for their daily computing operations [28]. The infrastructure that runs these computing services is normally in different geographical areas from these organizations. It is managed by third party organizations and services are normally accessed through use of the internet. With this technology, called cloud computing, organizations shift their focus from costly construction, maintenance and administration of datacentres [6]. They concentrate on core business objectives of organizations and leave all Information Technology (IT) related matters to CSPs.

There are a number of definitions for cloud computing and among others are those shown below:

"The new paradigm that delivers on-demand pay-as-you-use computing services through use of the internet" [6].

"The utility computing model whereby cloud users pay for services based on their consumption" [19].

"A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet" [19].

NIST in the Special Publication 800-145 also explain cloud computing as *"a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction" [43].*

These definitions show that cloud computing is a utility whereby cloud users pay for only those services delivered to them. That is, users are thus billed in accordance to computing services provided and their time of usage. The definitions further explain that cloud computing services are delivered upon request and can be released immediately once required tasks are completed. According to article [28], client computers participating in this technology communicate with multiple inter-networked servers that are spread in different geographical areas. These servers exchange information among themselves for redundancy and business continuity. They use web service technologies to allow ease of access by client computers. Cloud computing also takes advantage of other technologies such as virtualization together with grid computing to provide elastic, on-demand and reliable computing resources [10].

By virtue of being an on-demand utility, cloud computing provides different service models to paying customers. The next section discusses these cloud computing service models.

2.2 Cloud Computing Service Models

There are different types of service models which differ based on the kind of service they provide. Most of the research efforts that discuss cloud computing, which include *NIST definition of cloud computing* [43], concentrate on three of these service models. These are: *Software as a Service, Platform as a Service* and *Infrastructure as a Service.*

2.2.1 Software as a Service (SaaS)

This service model provides software and their functionality as a service whereby ownership of software and their possessions are separated [21]. These software are deployed and managed by CSPs on behalf of end users. They are accessed by end devices through some interfaces such as web browsers and APIs. CSPs fully manage the underlying infrastructure, which include server hardware, operating systems, storage devices and network. According to article [73], SaaS is adopted from Application Service Provider (ASP) model which emerged in the 1990s. In this model, application capabilities are delivered as a service by third party vendors [52]. They are accessed remotely through use of the internet. These services allowed full customization of applications installed in vendor servers. This, however, led to inefficiency of applications which later resulted in extinction of the model. The re-invention of the model as SaaS was introduced in the 2000s. This model differs from the original model in that; 1) it does not allow full customization, and 2) it allows integration with other applications [73].

As it is mentioned, SaaS cloud infrastructure is fully deployed and managed by CSPs [43]. This provides a security advantage to the cloud infrastructure. It limits the chances of malicious end-users to deploying their own malicious software. The disadvantage of this service model is that it does not allow customization of applications to fully satisfy user needs.

2.2.2 Platform as a Service (PaaS)

In this service model, cloud users are provided with a computing environment which has enough resources to allow deployment of own applications [66]. The provisioned environment include, among others, hardware plus basic software such as operating systems

[8]. It also includes other tools such as middleware and databases. This kind of service model offers an ideal computing environment for developers who intend to focus primarily on software development. It helps them to avoid administration and maintenance overheads of underlying infrastructure [22]. According to Mitchell [45], PaaS provides essential platform for software development lifecycle, which include, software design, development, testing and deployment. PaaS is also viewed as the fundamental platform that allows development of SaaS model functionalities [22].

In comparison to SaaS service model, PaaS provides flexibility for end-users to deploy their own software. Although this is important to allow customized applications that satisfy user needs, it also introduces some security risks. Malicious users take advantage of this and deploy their own malicious software tools. Consequently, they are able to attack both co-located applications within the cloud infrastructure and also external computing platforms.

2.2.3 Infrastructure as a Service (IaaS)

This is the cloud computing service model that provides customers with fundamental computing resources such as processing, storage, networks and middleware [63]. It provides flexibility for customers to deploy and manage their own software instances, starting with the operating systems. These instances are separated logically by use of software and share physical resources [51]. The main role of CSPs in this model is to ensure that cloud datacentres are operational and resources are accessed [8]. They provide little administrative services such as housing, hardware and its maintenance. IaaS service model is the underlying platform that allows deployment of PaaS service model. This makes it the base for deployment of both SaaS and PaaS service models. Based on this, IaaS service model inherits all the security risks incurred in these two other service models. More so, it incurs additional security risks caused by co-location of user owned VMs. Malicious users in this service model are able to co-locate their VMs with their target VMs [25]. Once co-located, inter-VM attacks are achieved by using shared resources as channels to either disrupt, corrupt or spy these co-located targets. Due to higher possibilities of inter-VM attacks, IaaS is the service model of interest in this study. It is the service model referred to whenever ‘cloud’ is mentioned in this dissertation.

In addition to these service models are deployment models available in cloud computing. These are representations of cloud environments which are notable based on ownership [4]. The next section discusses these deployment models in detail.

2.3 Cloud Computing Deployment Models

Some different research efforts, which include *NIST definition of cloud computing* [43], categorizes cloud computing deployment models into four types. These are: *private cloud, public cloud, community cloud* and *hybrid cloud*. The study at hand focuses only on IaaS service model when discussing these deployment models, unless stated otherwise.

2.3.1 Private cloud

In this deployment model, cloud infrastructure is run for a single organization [43]. It is either hosted internally within the organization premises or externally. This model differs from traditional in-house datacentres in that its infrastructure is managed by third party CSPs. They use virtualization to provide scalable, on-demand and elastic cloud computing services [66]. The nature of this deployment model does not allow multi-tenancy. This therefore minimizes possible inter-VM security threats which are common in other cloud computing deployment models. There are, however, other possible threats that are brought by unwanted co-location of VMs in this model. For example, it is possible for external malicious users to use traditional attacks to compromise low level security VMs [76]. Once compromised, these malicious users take advantage of shared resources to attack co-located VMs with higher security levels.

2.3.2 Public cloud

In this deployment model, provision of computing services is open to general public [43]. It is not delivered to a single organization as it is the case with private cloud. Rather, multiple organizations and/or individuals hire computing services which are hosted on infrastructure owned and administered by third party CSPs. These computing services are delivered through use of public networks and provided upon user request. In this deployment model, there is high possibility of multi-tenancy within the cloud infrastructure. Unfortunately, this multi-tenancy introduces some architectural vulnerabilities brought by, for example, possible co-location of VMs belonging to adversary users [65]. In this case, there are possibilities of malicious activities that either spy, corrupt or disrupt other co-located VMs [15]. These acts take advantage of shared physical resources common in cloud architectures. Due to these, the study at hand takes public clouds as the most insecure deployment model.

2.3.3 Community cloud

In this deployment model, CSPs provide cloud computing services to a limited number of organizations [66]. Normally, they provide these services to organizations with common computing concerns such as security, compliance and jurisdiction. As opposed to public cloud, this deployment model does not deliver services through public networks and is not open to general public. This deployment model is similar to private cloud but with multiple organizations to provide services to. Based on this, it inherits possible security risks that are common in private cloud models. Although limited, there are also possibilities of security risks brought by multi-tenancy. These security risks are bound by the fact that community cloud serves fewer number of known clients. It is therefore easier to trace malicious activities within the cloud infrastructure.

2.3.4 Hybrid cloud

In this deployment model, a single organization uses a combination of discussed deployment models to achieve its business objectives [43]. Here, computing services are normally provided by different CSPs, with each providing a different deployment model. For example, an organization can use private cloud to store its sensitive data. At the same time, use applications provided by public SaaS cloud to perform computations on that data. The security risks encountered in this deployment model depend on the combination of deployment models used. It inherits the risks of all combined deployment models.

2.4 Cloud Computing Characteristics

The popularity of cloud computing over the past years is brought mainly by its characteristics. These characteristics describe structures and capabilities of the cloud infrastructure [4]. There are different characteristic models provided in published research efforts. For example, Mell and Grance [43] in the '*NIST definition of cloud computing*' describe cloud computing using five characteristics: 1) *on-demand self-service*, 2) *broad network access*, 3) *resource pooling*, 4) *rapid elasticity*, and 5) *measured services.* Ali et al. [4] use the same characteristic model, but with an additional characteristic called; 6) *multi-tenancy.* For this study, there is a characteristic model that formulates definition of cloud computing. This model strives to define cloud computing architecture based on the theme of this research. It provides an overview of cloud computing architecture and how it promotes insecurities brought by placement of VMs within its infrastructure. The model is used throughout this study in order to tackle the problem at hand. The cloud computing characteristics used in this study are discussed next:

2.4.1 On-demand self service

This is common to both the discussed characteristic models provided in articles [4] and [43]. It means that computing services are provisioned to cloud users upon request, and without requiring any human interaction. The process is automated and therefore limits cloud users' choice on placement of their own VMs. This promotes chances of unwanted co-locations of VMs that lead to higher security risks.

2.4.2 Utility

This characteristic shows that cloud services are provisioned based on user demands, and charged in accordance to their usage rates [19]. It is equivalent to '*measured service*' in the discussed characteristic models provided in articles [4] and [43]. Using this characteristic, there are possibilities of anonymously hiring VMs for malicious intends. These are released immediately once their tasks are completed, leaving no trace of their owners.

2.4.3 Aggregate

This characteristic shows that cloud computing consolidates multiple instances of operating systems within the same PMs to minimize operational costs [60]. These instances are isolated using virtualization software that provides each with an illusion that it is running in its own PM. They share physical computing resources such as memory, storage, network and CPU. This characteristic is almost equivalent to '*multi-tenancy*' in characteristics provided in article [4]. The difference between the two characteristics is that *aggregate* also includes cases of private clouds, where consolidated VMs belong to a single client. That is, there is no multi-tenancy in private cloud deployment model. However, VMs belonging to same user are aggregated within same PMs. This characteristic promotes possibilities of attacks among co-located VMs. They take advantage of shared resources to either monitor, disrupt or harvest sensitive information that is used for malicious intends [76].

2.4.4 Self-managing

This is a newly introduced characteristic unique to this study. It shows capabilities of the cloud to automatically react to situations that lead to unwanted circumstances. These circumstances include those that put in jeopardy optimization and/or security objectives observed. Self-managing characteristic allows automatic assignment and reassignment of VMs within the cloud in order to minimize these unwanted circumstances [4]. These include, among others, issues such as overloading, faults and security breaches.

Self-managing characteristic includes within it, '*rapid elasticity*' characteristic discussed in the two characteristic models provided in articles [4] and [43]. This '*rapid elasticity*' is the automatic assignment and reassignment of VMs within the cloud infrastructure to minimize under-utilization or over-utilization of resources [46]. It is reaction of the cloud to minimize conditions that put in jeopardy optimization objectives. It however, does not take into considering security objectives.

2.4.5 Broad network access

This is also common to all the discussed characteristic models provided in articles [4] and [43]. It shows that cloud resources are accessed remotely through use of networks. This makes malicious activities common in traditional computing such as SQL injections to be possible in clouds.

2.4.6 Shared resources

This characteristic is almost equivalent to '*resource pooling*' discussed in the characteristic models provided in articles [4] and [43]. In addition to seemingly unlimited pool of resources shown in *resource pooling*, it specifies that these resources are shared among instances of operating systems. This means, although they are logically separated, instances access the same physical resources which include CPU, memory, storage and network. Malicious users take advantage of these shared resources to launch inter-VM attacks such as side channels [76].

Comparison of the discussed characteristic models is tabulated using Table 2.1 below.

Table 2.1: Comparing Cloud Computing Characteristic Models

New Model	**First Model**	**Second Model**
On-demand self service	on-demand self service	on-demand self service
Utility	measured service	measured service
Aggregate		(multi-tenancy)
Self-managing	(rapid elasticity)	(rapid elasticity)
Broad network access	broad network access	broad network access
Shared resources	(resource pooling)	(resource pooling)

In this table, **New Model** refers to the proposed model for the study at hand, **First Model** is the characteristic model provided by Mell and Grance [43], and **Second Model** refers to the characteristic model provided by Ali et al. [4]. The characteristics shown in brackets are those that are inclusive in the definitions of the corresponding characteristics shown under **New Model** column.

Based on these characteristics, the author of this research defines cloud computing as:

> *"an on-demand self-service and self-managing computing utility that aggregates multiple instances of operating systems that share computing resources, and accessed through use of the internet"*

Of the mentioned technologies used in cloud computing, *virtualization* plays a vital role in attaining cloud computing characteristics discussed. To understand *virtualization* better, the next section discusses it in detail.

2.5 Virtualization

Virtualization is an old technology which dates back to 1960s. It was first used in mainframes to logically partition their physical resources to allow sharing by multiple applications [53]. Recently, virtualization is broadened to allow multiple virtual instances of operating systems to run within the same PMs [14]. It achieves this by creation of virtual environment with multiple 'containers' that mimic a particular computing platform [63]. These containers are independent and isolated from each other, with each hosting its own instance of operating system. The isolation is, however, only logical because these instances share computing physical resources [14]. Virtualization provides management capabilities to these hosted virtual instances throughout their life-cycles [10]. These management capabilities include, among others; creation, deletion, suspension and migration of these virtual instances.

In order to provide support to cloud computing characteristics discussed, virtualization uses a type of software called ***hypervisor*** [2]. The next section discusses this software in detail.

2.6 Hypervisor

This is a virtualization software that lies between physical hardware of the PM and hosted VMs [49]. It provides VMs with an illusion that each is hosted in its own PM.

According to Reuben et al. [53], hypervisor provides two main benefits to cloud computing architecture, which are *isolation* and *resource sharing*. This study introduces the third benefit, which is *VM management capabilities*. This is introduced because two benefits provided by Reuben et al. [53] only support two of the discussed cloud computing characteristics. The supported characteristics are aggregate and resource sharing. The newly introduced benefit, therefore, provides support to the remaining characteristics that are not supported by *isolation* and *resource sharing*.

There are, therefore, three main benefits discussed in this research. These are: *1) Isolation, 2) resource sharing,* and *3) VM management capabilities*. These three benefits, together with characteristics they support, are discussed in the next subsections.

2.6.1 Isolation

Isolation is referred to as the complete separation of VMs that are running in the same PMs [53]. These VMs are isolated from each other and also from the operating system running in host PM. That is, programs running in one VM cannot be seen by other programs running in another. In order to achieve full isolation between co-located VMs, a combination of the following is used: fault isolation, resource isolation and security isolation [60]. The authors of the last cited article define fault isolation as the ability of the hypervisor to avoid sharing of code between co-located VMs. This helps to avoid exchange of codes with bugs which might result in disruption of operations of other co-located VMs. They further define resource isolation as the ability of the hypervisor to enforce preservation of fair resource sharing between co-located VMs. This isolation seeks to avoid undesired interactions between co-located VMs, known as crosstalk, which take advantage of shared resources. They lastly define security isolation as the ability of the hypervisor to limit access to logical objects in order to avoid conflicts of global resources or data. These global resources and data include, among others: files, virtual memory addresses, and process ids. Isolation plays a vital role in one of the cloud computing characteristics discussed in this study, which is *aggregate*.

2.6.2 Resource Sharing

Resource sharing is referred to as shared use of physical resources of underlying hardware [53]. Hypervisors provide VMs with virtual resources such as virtual memory, virtual storage, and virtual Network Interface Cards (NICs). These virtual resources access shared physical resources of host PM in a scheduled manner for processing the needs of hosted VMs [49]. The shared physical resources include, among others, CPU, memory, network resources and storage. Resource sharing plays a vital role in one of the cloud computing characteristics discussed earlier, and that is *shared resources*.

2.6.3 VM Management Capabilities

This third main benefit refers to overall management of hosted VMs throughout their life cycles. It provides support to the discussed cloud computing characteristics that are not supported by first two benefits. Firstly, it supports *on-demand self-service* characteristic by providing control functionalities to hosted VMs [32]. These control functionalities include automated creation, deletion and cloning of these VMs based on user demands. In addition, hypervisors also provide metering capabilities throughout lifecycle of created VMs [43]. They help support *utility* characteristic that enables transparent billing of users based on resource usages. Secondly, hypervisors also support *self-managing* capabilities by allowing automatic migration and/or suspension of hosted VMs [4]. This allows safeguarding against unwanted circumstances predefined in SLA that might jeopardize availability, integrity, confidentiality and/or performance of hosted VMs. Lastly, hypervisors support *broad network access* characteristic of cloud computing. They achieve this by providing I/O and networking interfaces that enable controlled access and communication with hosted VMs [49].

The hypervisor benefits discussed, together with cloud computing characteristics they support, are summarized by author of this research using Table 2.2 below. Each of the cloud computing characteristics discussed in this study is classified under hypervisor benefit that supports it.

Table 2.2: Main Benefits of the Hypervisor

Isolation	**Resource Sharing**	**VM Management Capabilities**
Aggregate	Resource sharing	On-demand self-service
		Utility
		Self-managing
		Broad network access

2.7 Conclusion

Hypervisors play a significant role in cloud computing. They provide characteristics that define behaviour of this paradigm. Among others, they provide full isolation between VMs that reside within same PMs. This isolation is brought by virtualization technology which allows sharing of physical resources. However, the isolation is only logical and can easily be broken [53]. The breakage takes advantage of shared physical resources that are used as channels to launch malicious inter-VM attacks such as side-channels [76]. This means that, co-location of VMs brought by use of hypervisors in cloud computing introduces some architectural vulnerabilities. The intensity of these vulnerabilities differ depending on the service model and/or deployment model used in the cloud architecture. There is therefore the need to consider security in cloud computing in order to minimize these vulnerabilities. The next chapter is an introduction to cloud computing security.

Chapter 3

Cloud Computing Security

Cloud computing introduces a number of security benefits that circumvent risks common to traditional computing. These security benefits include, among others, centralization of security, process segmentation, redundancy and high availability [78]. Although the number of common security risks are countered creditably in cloud computing, there are, however, other security challenges that are introduced by the cloud architecture [54]. These challenges are associated with cloud computing characteristics that are discussed in previous chapters. If not countered for, the cloud computing specific challenges result in security risks that jeopardize integrity, availability and confidentiality of hosted VMs.

This research focuses on the security aspect of cloud computing that strives to minimize vulnerabilities introduced by undesired co-location of VMs. More specifically it concentrates on vulnerabilities brought by undesired co-location of VMs belonging to adversary users, which normally promote *inter-VM attacks*. These inter-VM attacks, which are security risks of focus in this study, are simply malicious activities that strive to disrupt, corrupt or spy co-located VMs. They are discussed in detail in the next section to provide an overview of how they happen. Other sections discuss different cloud computing security challenges that promote these inter-VM attacks.

3.1 Inter-VM Attacks

In cloud computing, *aggregate* and *shared resources* characteristics make it possible for VMs belonging to different users to reside within the same PMs and share computing resources. In this case, there are possibilities of VMs belonging to users who are not of trust to each other to reside within the same PMs [4]. Even worse, there are also possibilities of VMs belonging to malicious users to reside within the same PMs with their targets. These possibilities are referred to by this study as *undesired co-locations* of VMs. They are promoted by placement of VMs that do not take into consideration security as one of their objectives. If not countered for, undesired co-locations normally result in security breaches that expose to danger confidentiality, integrity and availability of hosted VMs. There are different kinds of inter-VM attacks which include, among others: *cross-VM side channels attacks*, *cross-tenant attacks* and *covert channels attacks*.

3.1.1 Cross-VM side channels attacks

These are inter-VM attacks that take advantage of known vulnerabilities to attack co-located VMs [76]. According to the last cited article, VM images that are normally used in the cloud have known vulnerabilities. Malicious activists take advantage of these known vulnerabilities to launch inter-VM attacks. For example, image publishers normally leave behind sensitive information such as passwords, crypto-keys and other credentials [12]. Scanning these images may result in acquisition of these information. Once acquired, this information is used to gain access to target VMs to either corrupt, disrupt or spy on them. Other examples of cross-VM side channels include those provided by Afoulki et al. [2]. They include ability to exploit known vulnerabilities of hypervisors to attack co-located VMs. Once exploited, hypervisor provides privileged access that allows ease to disrupt, corrupt or spy hosted VMs. Another example is the exploitation of some drivers used within PMs [2]. This exploitation allows malicious users to directly access underlying hardware such as physical memory. This provides these malicious users unwanted access to hosted VMs through bypassing access control mechanisms. In general, cross-VM side channels can be explained as the type of inter-VM attacks that take advantage of known vulnerabilities within the PM. They do not use brute-force to gain unwanted access to target VMs.

3.1.2 Cross-tenant attacks

These are inter-VM attacks which are promoted by the fact that cloud users, more especially in IaaS cloud, are granted super-user rights [4]. This means that users with malicious intent are able to acquire IP and MAC addresses of other co-located VMs. Having this information together with super-user access rights, malicious users can launch attacks such as sniffing over networks.

3.1.3 Covert channels attacks

Covert channels, which are also common in traditional computing, are possible in virtualized environments. These were first explained by Lampson [36] as malicious activities that allow transfer of information between processes that are not allowed to communicate. In cloud computing, there is also a possibility of creating unwanted communications between co-located VMs [15]. These unwanted communications take advantage of shared resources to transfer unwanted programs that might endanger security of co-located VMs.

Having discussed inter-VM attacks, the next step is to explore cloud computing challenges that promote these kinds of attacks. The next section is an overview of cloud computing security challenges.

3.2 Cloud Computing Security Challenges

This study focuses on security challenges brought by structure or design of the cloud architecture. More specifically, it concentrates on challenges brought by vulnerabilities within the hypervisor. As mentioned, the hypervisor is a virtualization software that makes it possible to achieve, among other cloud computing characteristics, *aggregate* and *shared resources*. With these characteristics, the hypervisor introduces some cloud computing specific security challenges. These challenges are promoted by, for example, *consolidation of multiple systems (VMs)*, *sharing of physical computing resources*, and *complex code of the hypervisor.*

There are other security challenges available in cloud computing that are not necessarily the results of vulnerabilities within the hypervisor. These are security challenges introduced by other components of the cloud architecture such as shared *communication channels*. If not countered for, these kind of security challenges also result in inter-VM attacks. In addition to these are non-technical challenges such as those brought by *contractual and legal* issues between CSPs and cloud users. These kind of security challenges also have possibilities of causing malicious harm if not addressed appropriately. They are, together with other security challenges mentioned in this section, discussed in detail in the next subsections.

3.2.1 Consolidation of Multiple Systems

One of cloud computing characteristics that is discussed in this study is *aggregate*. As mentioned, it is consolidation of multiple VMs into same PMs in order to minimize operational costs while increasing utilization. Despite the mentioned advantages, aggregate is associated with some security challenges. This is because each of the consolidated VMs is associated with known vulnerabilities [53]. According to Yuchi et al. [76], introduction of a new VM image of known vulnerabilities impose some security risks to co-located VMs. With these known vulnerabilities, malicious users exploit VMs and use common inter-VM attacks such as cross-VM side channels to attack co-located VMs. To support this, Ali et al. [4] state that in a virtualized environment, security level of the entire system is the same level as security of the weakest VM. Having a number of VMs with known vulnerabilities consolidated within the same PM therefore means increased points of attacks to the virtual environment.

To counter for these kinds of vulnerabilities, Yuchi et al. [76] introduced a security-aware VM placement algorithm. This algorithm allocates VMs based on their vulnerability statuses together with survivability statuses of PMs. It places VMs with low risk probability into PMs with high survivability and vice versa. To calculate survivability status of PM, firstly, the vulnerability status of hosted VMs are quantified. This is based on vulnerability scores obtained from National Vulnerability Database (NVD). Secondly, the risk probability for each VM is explored based on dependency relations with other

VMs in the cloud. Lastly, the security score for each PM is calculated based on vulnerability status plus risk probabilities of hosted VMs. After these steps, then it is easier to place the newly requested VMs based on their vulnerability status and survivability score of available PMs.

Although the discussed solution [76] minimizes possible threats brought by co-location of VMs, it however, does not factor out possible threats brought by co-location of VMs belonging to adversary users. These kinds of threats are common in cloud service models that allow multi-tenancy. In addition, this solution focuses solely on security aspect and ignores optimization objectives that minimize costs of running cloud datacentres and ensure good QoS. The advantage of this solution is that it does not require a change in the cloud architecture. It simply modifies a component of the hypervisor that allocates VMs to available PMs.

3.2.2 Sharing of Physical Computing Resources

One of the main challenges in cloud computing is ensuring that there is no interference between co-located VMs that share resources [60]. The situation even becomes worse in multi-tenant cloud computing models. Here, malicious users take advantage of shared physical resources to disrupt, corrupt or spy other co-located VMs [15]. For example, it is possible for malicious users to take advantage of shared resources and use cache read operations to obtain sensitive information such as cryptographic keys from shared cache memory [25]. In aggregate environment, it is also possible for malicious users to exhaust shared physical resources to cause Denial of Services (DoS) to co-located VMs [4]. In order to successfully co-locate their VMs with their targets, malicious users normally use brute-force strategies [26]. Sometimes they take advantage of sequential and parallel locality common in most VM placement strategies [26].

There are proposed countermeasures that minimize attacks brought by undesired co-locations such as those that concentrate on secure placement of VMs. They include work by Al-Haj et al. [3] which proposes a security-aware resource allocation based on security groups created from input data. This input data is a collection of the following: cloud provider's constraints, user's resource requirements, user's security requirements and user's risk metrics. The provider's constraints include capacities and availability of computing resources, topology information and cloud management thresholds. The user's resource requirements include number of VMs required and their resource requirements such as CPU, memory, and storage capacity. The user's security requirements include adversary user lists and security/risk enforcement specifications. The risk metrics include vulnerability scores for VMs, which are obtained from NVD. The resource allocation and overall placement of VMs is based on the combination of these input data. Closely related is work by Afoulki et al. [2] where they also propose a security-aware placement for VMs in IaaS cloud. In this article, authors strive to reduce risks brought by co-location of VMs belonging to adversary users. They replace the scheduler within Opennebula VMM with an algorithm that checks for adversary users before placement. For cases where all available PMs are occupied by adversary users, this algorithm takes into consideration migrations of VMs belonging to adversary users in order to satisfy newly requested VMs. The advantage of these proposed countermeasures is that they do not need modifications to the cloud architecture. Their shortcoming is that they concentrate solely on security aspect and ignore optimization principles.

3.2.3 Complex Code of the Hypervisor

Different from the discussed security challenges are those introduced by vulnerabilities within the hypervisor source-code itself. The hypervisor is a bulky piece of software with approximately 100,000 lines of code [49]. Like all software, it is associated with vulnerabilities or bugs introduced in its development lifecycle [34]. Its large size of code contribute to the likeliness to have these bugs. There are known vulnerabilities associated with different commercial hypervisors. These vulnerabilities are documented in Common Vulnerabilities and Exposure (CVE) databases [65]. For example, in September 2017 CVE reports documented 234th (CVE-2017-14319, xsa234) vulnerability identified for all versions of XEN hypervisors [47]. If not sufficiently patched, malicious users take advantage of these known vulnerabilities to gain privileged access to the cloud. They use, among other attacks, some common classical software exploits such as buffer overflows to compromise hypervisors and insert some malicious pieces of code [2]. Once compromised, they can use these privileged access rights to jeopardize the integrity, availability and confidentiality of hosted VMs [33]. In addition, hypervisors also have some commercial consoles that are used for administration purposes [72]. These consoles are off-the-shelf text entry and display devices used to remotely manage virtual environments. They add complexity to cloud architecture which results in increased attack base. The possible attacks that take advantage of these consoles include, among others SQL injections and Cross-site scripting [72].

There are proposed countermeasures against hypervisor attacks and vulnerabilities. Among them are those proposed by Wang et al. [68] where they introduce a model called HyperCheck. This model ensures security within the hypervisor by monitoring CPU and memory of host machine. It has an analysis module that reads and analyses memory contents based on three properties: linearity, stability and perpetuity. This model further takes screenshots of hypervisors and compares with the initial to check whether there are any malicious activities that tried to compromise the hypervisor. The main aim of this model is to ensure the integrity of the hypervisor and data within the virtualized environment. Wu et al. [71] also proposed a system that reduces attack base of the hypervisor by dividing it into smaller modules. Most of these modules are decoupled from the running host operating system and are executed in less privileged (user) mode. Szefer et al. [64] implement a prototype that eliminates use of the hypervisor in order to avoid threats associated with hypervisor vulnerabilities. This prototype uses extended capabilities of hardware such as hardware paging mechanisms and virtualized I/O devices to host and manage multiple VMs within the PM.

A common problem with the cited countermeasures above is that they all require a change to the cloud architecture which makes them costly. These changes include use of analysis modules that read and analyze memory contents, as proposed by Wang et al. [68]. Furthermore, these countermeasures are beyond the scope of this study as the primary focus is on reducing risks brought by undesired co-locations of VMs.

3.2.4 Communication Channels

Communication channels add more security challenges to the cloud infrastructure. These challenges are brought by vulnerabilities in data transmission channels of cloud infrastructures. They involve both *internal communication challenges* and *external communication challenges*. These security challenges are discussed next.

Internal Communication Challenges

Associated with *shared resources* characteristic are challenges brought by internal communication channels. These are channels within the cloud infrastructure that transmit data between VMs. They also provide communication between VMs and storage devices. Among vulnerabilities introduced by these internal communication channels are those brought by *shared internal communication channels* and *virtual networks*.

Shared Communication Channels Vulnerabilities: The distributed nature of cloud infrastructure together with massive resource sharing and VM instance synchronization generate a lot of data in-transit [23]. This data in-transit share physical network infrastructure within the cloud architecture [4]. The most common challenges brought by this sharing of network infrastructure components are cross-tenant attacks [27].

Virtual Networks Vulnerabilities: In cloud computing, network virtualization is used to isolate user network traffic in order to provide scalability, flexibility and basic network security [30]. The virtual networks which are independent of underlying physical networks are created within shared network resources. They are normally configured in different hypervisors using either one of two modes: *bridge* or *route* mode. According to Wu et al. [72], in bridge mode, interfaces of VMs are directly attached to software bridge which is connected to physical network. This bridge acts as a hub and broadcasts all traffic to directly connected interfaces of VMs. These broadcasts impose some vulnerabilities to the cloud infrastructure. That is, it is possible for directly connected VMs to sniff traffic within network using tools such as Wireshark. In route mode, the administrator creates a point-to-point means of communication between VMs using a virtual switch [72]. This virtual switch uses some IP/MAC tuples for direct communication with dedicated VMs. Attackers in this case spoof the Address Resolution Protocols (ARPs) and redirect traffic to their own VMs.

There are proposed solutions to counter for above mentioned communication challenges. To discuss a few, Wu et al. [72] suggested a layered model that combines characteristics of the *route* and *bridge* modes to develop a more secure environment. This model consists of three layers which are: routing layer, firewall and shared network. The routing layer is responsible for communication between physical network and virtual network. It has the same functions as traditional route mode. The firewall prevents spoofing attacks within shared networks. It achieves this preventions by dropping packets within virtual network that try to modify routing table. The shared network layer separates communication networks between untrustworthy VMs. This means, VMs that do not have authorized communication are allocated different virtual networks. But, those that communicate with each other are allocated shared network. Komu et al. [33] also proposed use of Host Identity Protocol (HIP) to cryptographically authenticate communication between VMs. This is to limit inter-VM attacks such as impersonation and data interceptions. It is ideal for communicating applications with multiple components that are spread across a number of VMs. The problems with cryptographic authentications within cloud networks are additional processing costs which results in higher energy consumption costs. In addition, solutions discussed [33, 72] are costly as they require change to the cloud architecture. Although these proposed solutions are important in securing the cloud, they are of little interest to this study. This is because they do not consider secure placement of VMs in order to minimize risks brought by stated vulnerabilities.

Conversely, there are other research efforts that consider secure placement in order to minimize risks brought by internal communication channels. These include work by Afoulki et al. [2] where authors propose VM placement based on internal network isolation. In this solution, virtual networks are used to isolate traffic between VMs belonging to untrusted users. This is achieved by grouping trusted VMs and assigning to particular virtual network. The virtual networks are spread across different PMs within the cloud infrastructure, therefore influencing the placement of VMs. Each VM is placed in a PM based on the group it is assigned to. The isolation requirements that help achieve grouping of VMs are specified in user policies. Like other discussed security-aware VM placements, the shortcoming of this solution is that it focusses solely on security and ignores optimization objectives. In addition, it does not factor out possibilities of inter-VM attacks that are promoted by undesired co-location of VMs.

External Communication Challenges

These kind of communication challenges are associated with *broad network access* characteristic of cloud computing. They involve challenges brought by vulnerabilities in external communication channels such as web services. These web services are channels used to remotely access and manage cloud computing services through use of browsers and APIs [4]. The use of these web services in cloud computing results in introduction of vulnerabilities common to traditional web applications [4]. These are vulnerabilities that promote common attacks such as; SQL injections, cross-site scripting, sensitive data exposure, and insecure direct object references. Although these vulnerabilities are the same for both cloud and traditional web applications, countermeasures to these are different. This is because of the complex nature of cloud architecture which include, among others, characteristics such as *aggregate* and *shared resources*.

As mentioned, web services use APIs as one of the tools that provide access to cloud computing resources. Due to this, security levels of these APIs influence overall security of the cloud infrastructure. Consequently, low levels of security in APIs result in security breaches that affect availability of the cloud services. APIs describe features of the cloud architecture [4], as they are used to build and extend cloud services. CSPs normally publish their APIs as a means of marketing their cloud infrastructure to the outside world. Although this is used as the marketing strategy to attract end-users, it may also be used by malicious users to cause harm. Knowing details of cloud computing infrastructure makes it easier for attackers to use known vulnerabilities to either disrupt, corrupt or spy hosted VMs.

There are countermeasures proposed to minimize security risks brought by APIs. For example, Sirisha and Kumari. [58] introduced a two stage API access control mechanism in order to secure cloud APIs. They use a Role Based Access Control (RBAC) model to implement their security mechanism. This is an access control mechanism that grants permissions based on roles assigned to authorized users. To achieve their two stage APIs, this mechanism provides required access to authorized users. It firstly validates attributes of the user against the list of registered users in the database. These attributes include IP addresses and domain names of machines that access the cloud service. They help identify organization in which the user accessing cloud services belongs to. Secondly, the mechanism validates roles of the user against user-role-permission database. This is to determine resources or objects that the user is permitted to access. Although Sirisha and Kumari [58] provides secure access through use of APIs, there is, however, some limitation identified in their solution. The limitation is that access to the cloud services is limited to devices that are within the registered domain. There is no flexibility of accessing these services from anywhere and by using any end-device. The security of APIs is however beyond the scope of this study.

3.2.5 Legal and Contractual Challenges

These are non-technical challenges brought by conflicting legal issues. They are brought by violations of agreements between CSPs and cloud users [4]. To explain this further, cloud computing infrastructure is normally owned by third party CSPs that lease computing resources to paying cloud users. This means that cloud users have limited administrative control over their data and applications. They rely on CSPs for overall performance and security of their applications and data. In order to ensure satisfactory service delivery, the two parties sign a written agreement that states terms and conditions which are related to legalities, SLA and location of the data [4]. This agreement includes issues such as performance assurance, regulatory law enforcement, monitoring of contract enforcement and geographic jurisdictions. There are however some external factors that may result in breach of the agreements. These include the different laws at different geographical jurisdictions. They also include malicious attacks that compromise performance of the cloud.

SLA Challenges

SLA is a documented agreement on terms and conditions of service delivery between CSPs and end users [4]. These agreements include expected minimum performance level, user security requirements and penalty costs in cases of breach of SLA contract. Failure to sufficiently implement SLA results in security gaps [23]. For example, it is difficult to make claims to third party sub-contracted CSPs in cases of SLA breaches if issue of sub-contracted CSPs is not well addressed in the SLA [4].

Legal Jurisdiction Issues

These are non-technical issues that are brought by the distributed nature of the cloud infrastructure across different geographical areas with different legal jurisdictions [23]. In this case, it is difficult for cloud users to configure security policies that comply with legal jurisdictions of different geographical locations [4].

3.3 Summary

There are a number of vulnerabilities common in cloud computing that are brought by both technical and non-technical issues. For technical issues, there are vulnerabilities brought by complexities in the cloud architecture and communication channels. These complexities are the result of some cloud computing characteristics. They involve integration of VMs into the same PMs and sharing of physical resources. Here, malicious users take advantage of these complexities to launch inter-VM attacks. For non-technical issues, challenges involve breach of written agreements and legal contracts. These breaches normally lead to malicious or undesired circumstances that jeopardize integrity, availability and confidentiality of the cloud. The summary of the discussed cloud computing challenges is presented with an aid of a table shown in table 3.1. In this table, each security challenge is associated with cloud computing characteristic that promotes it.

To counter for these vulnerabilities, there are solutions proposed in different research works. Some of these solutions require a change in the cloud architecture, which is costly. However, there are other solutions that only modify the hypervisor without need to change cloud architecture. Most of these solutions replace hypervisor component known as a *VM scheduler or placement algorithm.* These are a set of instructions which selects appropriate PMs for VM placement based on their objectives. They are replaced with those that consider *secure* placement as part of their objectives. The idea being to minimize risks brought by mentioned vulnerabilities. These kind of solutions do not require any additional costs.

Table 3.1: Summarized Cloud Computing Security Challenges

Security Challenges / Vulnerabilities	**Possible Risks**	**Associated Characteristics**	**Countermeasure Approaches**
Consolidation of multiple systems	Inter-VM attacks	Aggregate	Secure VM Placement
Sharing of physical resources	Inter-VM attacks	Resource sharing	Secure VM placement
Complex code of the hypervisor	Classical software exploits to gain privileged access rights	Self-managing	Changing cloud architecture (e.g. eliminating use of the hypervisor)
Communication Channels (internal and external)	Inter-VM attacks Traditional Web application attacks (e.g. SQL injections)	Resource sharing Broad network access	Secure VM Placement Changing cloud architecture (e.g. implementing 2 stage API access control)
Legal and Contractual challenges	SLA violations Legal Compromise	Utility	Penalties and Fines Laws and Regulations

3.4 Conclusion

This chapter focusses on cloud computing security challenges. It stipulates different security challenges and vulnerabilities that promote them. Among the discussed vulnerabilities are those that result in inter-VM attacks. Minimizing these kind of attacks is the core interest of this study. There are some discussions on countermeasures against these cloud computing specific attacks. From these discussions, the most suitable countermeasures are *secure placement of VMs* within the cloud infrastructure. These are achieved through use of the hypervisor component called ***VM placement algorithm***. In order to achieve secure VM placement, security needs to be added as one of the objectives in the implementation of these VM placement algorithms. For better understanding of VM placement algorithms, the next chapter discusses them in detail.

Chapter 4

VM Placement Algorithms

The previous chapter (Chapter 3) discusses cloud computing security challenges and their possible countermeasures. Among discussed countermeasures are secure placement of VMs within the cloud infrastructure. These are used mainly to minimize risks brought by undesired co-locations of VMs in aggregated environments. The secure placement of VMs are achieved by use of hypervisor module known as *VM placement algorithm*. This is a set of instructions on how VMs should be assigned and/or reassigned to available PMs based on observed objectives. In order to understand VM placement algorithms, the next section discusses them in detail. These discussions include different categories of VM placement algorithms used in this study. For each category, examples of currently implemented VM placement algorithms are provided. These examples are further interpreted by author of this study using activity diagrams. The other section introduces Optimized Security-aware VM placement algorithm. It details the need for implementation of this kind of VM placement algorithm.

4.1 What are VM Placement Algorithms?

The process of selecting an appropriate PM with enough physical resources to host a particular PM is called virtual machine placement [40]. This process is conducted with an aid of algorithms called *VM placement algorithms*. The concept of VM placement algorithms originated in around 2007 where researchers found the need to automatically place VMs within virtualized environments [32]. This is to allow dynamic reallocation of VMs in order to manage physical resources. To successfully manage these resources, a hypervisor provides continuous monitoring services to physical resources [5]. In addition, it also provides runtime decisions to allow automated assignment and reassignment of VMs to accommodate changing workloads. The process of assigning and reassigning VMs to available PMs is a critical and complex process that is normally considered NP hard, more especially for large datacentres [42]. That is, there are m^n possible mappings for a datacentre consisting of m PMs with n VMs to be placed. In addition, unpredictable arrival of VM requests make the placement of VMs even more complex. This results in most of the VM placement algorithms being heuristics because it is difficult and time consuming to find accurate solutions.

VM placement algorithms are implemented to achieve one or more of particular objectives. The objectives include, among others; energy consumption minimization [38], traffic cost minimization [9], load balancing [77] and security-aware placement [15]. Based on these objectives, this study categorizes VM placement algorithms into three groups, which are: 1) VM placement algorithms that focus on QoS, 2) VM placement algorithms that focus on cost reduction, and 3) VM placement algorithms that focus on security. The next subsections discuss three categories of VM placement algorithms in detail.

4.1.1 VM Placement Algorithms that focus on good QoS

As mentioned, two characteristics of cloud computing, which are *aggregate* and *shared resources*, allow VMs to be consolidated within the same PMs and share physical computing resources. This is mainly to reduce running costs of cloud datacentres while utilizing physical computing resources. The consolidation of VMs is, however, limited by physical resource amounts available in PMs [35]. This means that each PM accommodates a finite number of VMs whose total resource requirements match its physical resource capacities. Based on this, there is a need to dynamically assign VMs to available PMs [5]. This dynamic assignment of VMs needs to ensure that VMs allocated to a PM have a fair share of its physical computing resources. Furthermore, it needs to ensure that physical resources available in PMs are not under-utilized. Failure to appropriately assign these VMs to available PMs results in a number of unwanted circumstances that normally affect QoS. Unwanted circumstances may include, heat imbalances in cloud datacentres [74], traffic bursts in datacentre networks [9], and migration overheads [41]. In order to minimize these unwanted circumstances, there are currently implemented VM placement algorithms that focus on good QoS. For example, some research efforts propose even distribution of VMs within the cloud infrastructure in order to minimize physical resource exhaustion which results in performance degradations [77]. Others propose frameworks that strive to detect and eliminate hotspots within cloud datacentres [74].

Among most cited VM placement algorithms that focus on QoS are those put forth by Biran et al. [9]. In their study, authors concentrate on efficient QoS delivery based on traffic demands of VMs. Their idea is to implement VM placement algorithm that absorbs traffic bursts caused by large variety of traffic patterns. They introduce a network-aware optimization solution called Min Cut Ratio-aware VM Placement (MCRVMP). This solution takes into consideration both resource constraints within PMs and network constraints. It strives to minimize traffic in the network cuts so that unpredictable traffic bursts can be absorbed. This MCRVMP is generally NP hard and therefore uses heuristic algorithm to solve at a reasonable time. This proposed VM placement algorithm has two main properties. Firstly it recursively traverse the tree network structure to place Connected Components (CCs) and VMs at one level of the tree at the time. Secondly it adopts two-phase approach that places CCs on the network and then expand them to place VMs on actual PMs. This VM placement algorithm (MCRVMP) which originates from work of Biran et al [9] is interpreted in this study by using an activity diagram shown in Figure 4.1.

To explain in details how MCRVMP algorithm [9] works, it firstly creates clusters of CCs, which are groups of communicating VMs. It then recursively place these CCs on different levels of the tree network starting at the root switch. This is followed by placement on other switches on next levels of the tree. Every switch in these different levels represent total capacity of the tree leave rooted at that switch. In this case, the switch that represent the tree leave is referred to as the Virtual Host (VH). For every placement of CCs, if the aggregated resource requirements of the CC_d (for $d = 1$ to

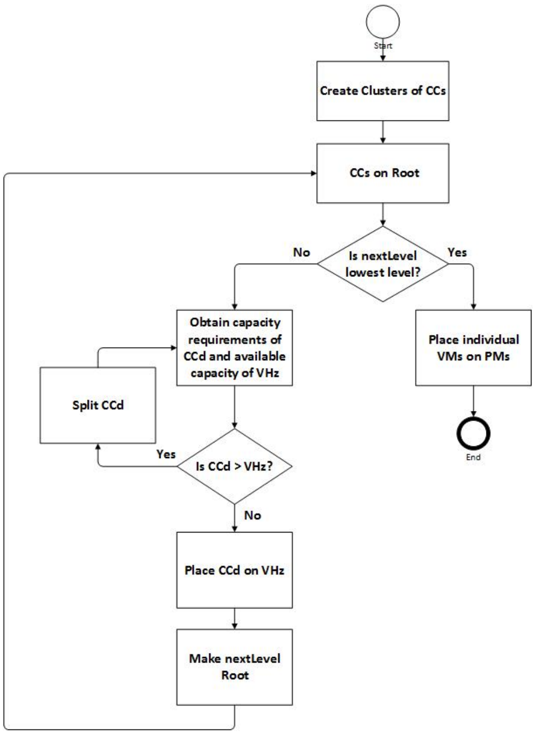

Start
Create Clusters of CCs
CCs on Root
Is nextLevel lowest level?
No
Yes
Obtain capacity requirements of CCd and available capacity of VHz
Place individual VMs on PMs
End
Split CCd
Yes
Is CCd > VHz?
No
Place CCd on VHz
Make nextLevel Root

Figure 4.1: Min Cut Ratio-aware VM Placement (MCRVMP)

n) exceed the aggregated capacity of VH_z (for $z = 1$ to m), the CC_d is split among subtrees of that switch. Again, if next level of the tree is the lowest level that comprises of PMs, then CCs are expanded and VMs are placed on actual PMs. Placement of CCs on VHs and expansion of CCs to place VMs on actual PMs at lowest level results in communicating VMs being placed in close proximity to each other. These communicating VMs are placed either within same PMs or in PMs that are within the same network leave. This results in minimization of total traffic costs within the cloud infrastructure. In addition, traffic flowing in each network cut becomes low compared to the total capacity of that network cut. The high traffic bursts are therefore easily absorbed at different network cuts. In summary, important point regarding MCRVMP is that it places the communicating VMs in close proximity to each other. This is to minimize possible network traffic bursts brought by high traffic within the cloud network. The problem with the proposed solution is that it works only in networks that are either tree networks or can be transformed into tree networks.

Other research efforts that focus on QoS and are similar to work proposed by Biran et al. [9] are those proposed by Meng et al. [44]. These research efforts also propose a VM placement algorithm that reduces traffic costs by placing communicating VMs into PMs that are in close proximity to each other. The additional objective observed in this VM placement algorithm is time to complete job. Vu and Hwang [67] also focus on placement of communicating VMs in PMs that are in close proximity to each other. The additional objective observed is minimization of energy consumption in cloud datacentres. All these discussed research efforts ([9], [44], and [67]) use connected components to reduce total traffic costs. The aim is to provide good QoS to paying customers at reasonable time. They, however, do not take into consideration security to minimize vulnerabilities brought by communication dependency of these connected components.

4.1.2 VM Placement Algorithms that focus on Cost Reduction

Cloud computing has received massive attention in recent years due to its ability to reduce datacentre running costs. This is achieved by use of virtualization to aggregate VMs into fewer number of PMs to reduce operational costs [9]. Nonetheless, there is normally a trade-off between performance and cost reduction in cloud computing [7]. This is because, aggressive consolidation of these VMs results in physical resource exhaustion that compromises application performances. There is therefore the need for extensive studies on placement of VMs within the cloud infrastructure to reduce running costs [32].

There are a number of currently implemented VM placement algorithms that focus on cost reduction. Among them are those that focus on energy consumption minimization to reduce costs. It is important to note that energy consumption costs are a concern in cloud computing. Recently, there has been a rapid increase in energy consumption by the cloud datacentres [48]. To support this, studies by Sverdlik [62] show that, energy consumed by datacentres in the US in 2014 is about 70 billion kilowatt-hour of electricity. This is equal to the amount of electricity consumed by about 6.4 million average homes in the US in that year. These figures double every year based on growth of the cloud datacentres. Because energy is a limited and costly resource, it is important, therefore, to consider VM placement algorithms that focus energy consumption minimization in order to reduce costs. One of the most cited studies that focus on placement of VMs to reduce energy consumption is that proposed by Yang et al. [75]. The proposed VM placement algorithm, known as *Sep-pack VM placement algorithm*, is interpreted in this study by using an activity diagram as shown in Figure 4.2.

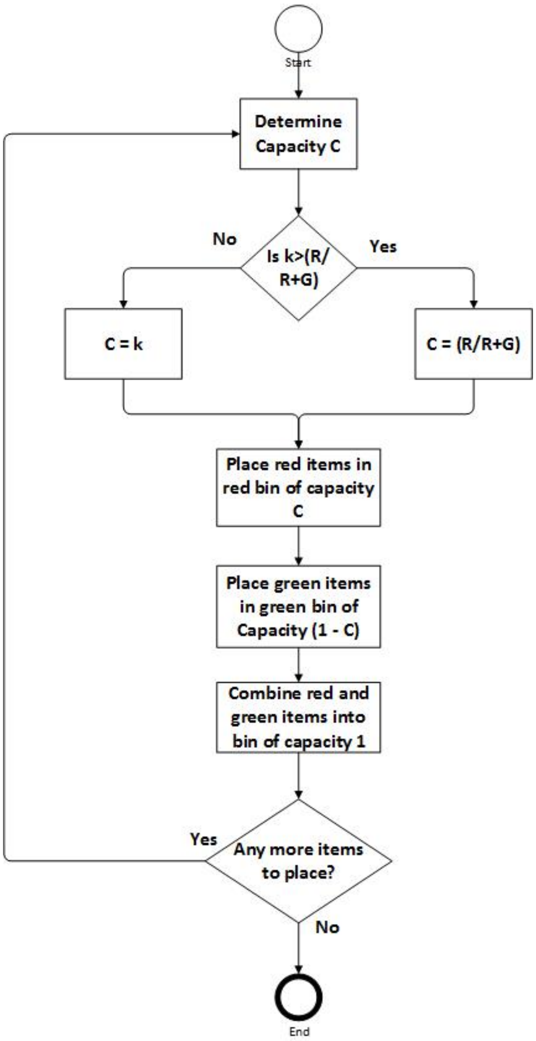

Start
Determine Capacity C
No
Is k>(R/ R+G)
Yes
C = k
C = (R/R+G)
Place red items in red bin of capacity C
Place green items in green bin of Capacity (1 - C)
Combine red and green items into bin of capacity 1
Yes
Any more items to place?
No
End

Figure 4.2: Sep-Pack VM Placement Algorithms

This algorithm shown in Figure 4.2 is an energy-aware VM placement algorithm with additional objective of optimizing application performance. In order to achieve its objectives, the study models its solution as a modified bin packing problem. Here, PMs are represented by bins and VMs are items. Furthermore, items are categorized into two, red and green items. The red items, on one hand, represent data-intensive VMs. The green items, on the other hand, represent CPU-intensive VMs. As a means to save energy, VM placement algorithm firstly places red items into bin of capacity C, where $C = \frac{R}{R+G}$ ($R =$ total weight of red items, $G =$ total weight of green items). This capacity C cannot exceed the maximum allowed total weight of red items k. Secondly, it places green items into bin of capacity $(1 - C)$. It then combines items in the two bins and consolidate them into the first bin of capacity 1. This results in fewer number of bins as each bin is utilized to its full capacity. In order to optimize application performance, this algorithm only consider green items (CPU-intensive) whenever there is a need for migrations. This is because migrating data-intensive VMs away from PMs that contain their images degrades their performance for up to 40 percent. It is therefore important to migrate only CPU-intensive VMs as there is an insignificant change to their performance if migrated away from their images.

In summary, VM placement algorithm proposed by Yang et al [75] reduces running costs by consolidating VMs into fewer PMs. In addition, it takes into consideration workload characteristics of VMs to optimize application performance. Although it strives to optimize performance, it has one factor that contributes towards application performance degradations. This is exhaustion of physical resources by consolidation of VMs to the full capacity of PMs. Also, the proposed algorithm does not take security as an additional objective. This means vulnerabilities brought by VM consolidation are not countered for in this proposed algorithm. There are therefore possibilities of inter-VM attacks brought by undesired co-locations. Other closely related studies are those proposed by Lin et al.[38], Ohta [48] and Kuo et al.[35]. All these studies consolidate VMs into fewer number PMs and switch off idle PMs to reduce energy consumption. They also do not take into consideration security aspect that reduces risks brought by consolidation of VMs.

4.1.3 VM Placement Algorithms that focus on Security

Despite financial advantage of sharing computing resources, cloud computing is associated with architectural vulnerabilities. As mentioned earlier, there are a number of factors that contribute towards these vulnerabilities. Among them are those brought by the discussed characteristics of clouds such as *aggregate* and *shared resources*. As mentioned, these characteristics allow consolidation of VMs into same PMs and allow sharing of physical resources. Malicious activities take advantage of shared physical resources to compromise integrity, confidentiality or availability of co-located VMs [25]. In order to minimize risks brought by these vulnerabilities, secure placement of VMs is considered. There are a number of currently implemented VM placement algorithms that focus on secure placement of VMs. One of the first and mostly cited security-aware VM placement algorithm is that proposed by Afoulki et al. [2]. In their work, they proposed a security-aware VM placement algorithm that avoids co-location of VMs belonging to adversary users.

To explain how this algorithm works, it starts by selecting a PM with enough physical resources to host a newly requested VM. Furthermore, it checks whether the selected PM hosts any VM belonging to adversary users. If there are no VMs belonging to adversary users then placement is completed, otherwise the algorithm searches for another compatible PM. For cases where all suitable PMs contain adversary users, this algorithm checks whether it is possible to suspend or migrate VMs belonging to adversary users in selected PM. The suspension depends on priorities of both VMs belonging to adversary users and newly requested VM. If neither suspensions nor migrations are possible, the algorithm checks if the newly requested VM can be scheduled for later placement, otherwise the VM request is declined. For demonstration purposes using activity diagrams, it is important to note that VMs that allow scheduling for later placement are referred to as Advanced Reservation leases (ARs). Those that require immediate placement are referred to as Best-Effort leases (BEs). In summary, the first prominent point in the proposed VM placement algorithms proposed by Afoulki et al. [2] is the consideration of adversary users in the placement of newly requested VMs. The second is the suspension and/or migration of VMs belonging to adversary users. This is possible only if there are no other suitable PMs to host the newly requested VM. The security-aware VM placement algorithm proposed by Afoulki et al. [2] is interpreted by the author of this study using activity diagrams in Figure 4.3.

Other closely related security-aware VM placement algorithm that minimizes co-location attacks by adversary users include that proposed by Han et al. [25]. This VM placement algorithm strives to minimize ability of malicious users to co-locate their

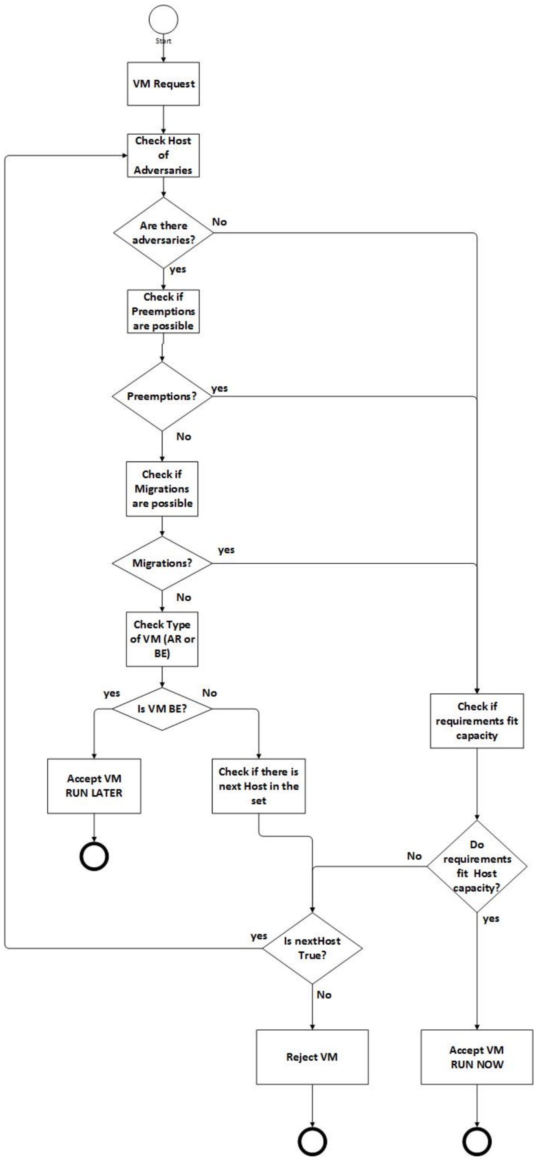

Start
VM Request
Check Host of Adversaries
Are there adversaries?
No
yes
Check if Preemptions are possible
Preemptions?
yes
No
Check if Migrations are possible
Migrations?
yes
No
Check Type of VM (AR or BE)
yes
No
Is VM BE?
Accept VM RUN LATER
Check if there is next Host in the set
Check if requirements fit capacity
Do requirements fit Host capacity?
No
yes
yes
Is nextHost True?
No
Reject VM
Accept VM RUN NOW

Figure 4.3: Security-aware Scheduler

VMs with their targets. They achieve this by using the policy called *Previously-selected-servers-first* for every user who requests a new VM. This minimizes spread of VMs belonging to the same malicious user across the entire cloud infrastructure. The proposed VM placement algorithm [25] does not, however, consider cases where malicious user requests new VMs using different identities. Different from this are a security-aware VM placement algorithm proposed by Carol et al. [15]. This algorithm takes user requirements as input to allow them to choose their required security levels [15]. Yuchi and Shetty [76] also propose a security-aware VM placement algorithm that separates more vulnerable VMs from those with lower vulnerability status. This minimizes chances of compromising vulnerability status of more secure VMs by taking advantage of co-location and shared resources.

4.2 Towards Optimized Security-aware VM Placement Algorithm

Although security in VM placement algorithms is of essence and is the theme of this research, other two categories discussed in the previous section are equally important. They allow provision of optimization objectives that bring about good QoS at reduced operational costs. There is, however, normally a trade-off between security and these other two categories. For example, the discussed security-aware VM placement algorithm proposed by Afoulki et al. [2] focuses solely on secure placement of VMs. It does not consider optimization objectives that provide good QoS at reduced costs. This might result in secure cloud environment but with poor service delivery to paying customers. In addition, costs of running this secure cloud environment might be unnecessarily high. In order to avoid this, the author of this study find it important to consider the notion of *Optimized Security-aware (O-sec) VM placement algorithm*. This is explained as combining into the same algorithm the objectives classified under VM placement algorithms that focus on security and those that focus on QoS and/or cost-reduction. The main aim of this study is, therefore, implementation of ***O-sec VM placement algorithm***. The approach towards implementation of O-sec VM placement algorithm is to use the already existing VM placement algorithm. This should be the algorithm that assumes higher security and/or optimization objectives as compared to other currently implemented VM placement algorithms. The qualifying VM placement algorithm will further be augmented towards O-sec VM placement algorithm.

4.2.1 Related Work

Ideally, security-aware VM placement algorithms require optimization objectives to reduce costs and provide good QoS. However, there is normally a trade-off between security objectives and these two mentioned optimization aspects. This results in most of the currently implemented security-aware VM placement algorithms observing only security objectives. Fortunately, recent research efforts strive to overcome trade-offs between the mentioned VM placement aspects. They introduce security-aware VM placement algorithms that take into consideration QoS and/or cost reduction objectives. For example, Gaggero and Caviglione [20] introduce a security-aware VM placement algorithm that observes cost reduction and QoS optimization objectives. In their work published in 2019, they strive to overcome possible hardware failures and unwanted reboots that might result in unavailability of VMs. This is achieved by use of Model Predictive Control (MPC) in placement of VMs. This model helps predict possible future hardware and software failures which can compromise availability of these VMs. To reduce costs, this algorithm consolidates VMs into fewer number of PMs to reduce energy consumption rates. Additionally, the algorithm ensures QoS by maintaining provision of services as stipulated in the SLA. This proposed VM placement algorithm is closely related to the outcome of this research. It is a security-aware VM placement algorithm that takes into consideration both QoS and cost reduction objectives. However, it does not factor out security risks of focus in this study, which are inter-VM attacks.

Closely related is work by Han et al. [26] published in 2017. In their work, they introduce a security-aware VM allocation policy that takes into consideration optimization objectives. This policy reduces possible inter-VM attacks that are brought by unwanted co-location of VMs. It is a modified version of a prototype called *"previous-selected-server-first"* (PSSF) which is proposed by the same authors [25]. They added energy consumption and workload balance to the security objective observed in PSSF. To ensure secure placement, PSSF allocates newly requested VMs to PMs with most VMs. The aim is to minimize strategies such as brute-force which are used by malicious users to co-locate their VMs with their targets. This modified version of PSSF [26] also combines security aspects with QoS and cost reduction optimization objectives. In addition, it minimizes possible inter-VM attacks brought by co-location with malicious VMs. However, it does consider possible co-location of adversary users. If not addressed, this might also result into inter-VM attacks.

Other related publications include work by Wong and Shen [70] published in 2018. In this work, they introduce a security-aware VM placement algorithm that takes into consideration QoS objectives. To provide secure placement, this algorithm minimizes possible co-location of malicious VMs with their targets. It achieves this by use of Familiarity concept to separate malicious VMs from their possible targets. This means, only those VMs that are 'familiar' to each other are co-located. The mentioned familiarity builds from previous co-locations that did not result in any malicious acts. To provide good QoS, this algorithm considers load balancing to avoid exhaustion of physical resources. This security-aware VM placement algorithm concentrates on QoS only. It ignores other optimization objectives that strive to minimize cost reduction.

4.3 Summary

In order to provide good QoS at reduced costs, and also to reduce risks brought by architectural vulnerabilities, cloud computing uses VM placement algorithms to dynamically assign VMs to available PMs. These dynamic assignment of VMs takes into consideration different objectives observed in the implementation of these VM placement algorithms. The research at hand categorizes VM placement algorithms into three groups, based on their objectives. The categories are:

VM placement algorithms that focus on good QoS: These are the kind of VM placement algorithms with the main objective of providing QoS stipulated in the SLA. They concentrate on objectives such as, time to complete job, optimal performances and reducing network bursts / traffic.

VM placement algorithms that focus on cost reduction: These kind of algorithms place VMs within the cloud infrastructure in a manner that reduces costs of running datacentres. They observe objectives such as, minimizing machine counts and minimizing energy consumption.

VM placement algorithms that focus on security: These kind of VM placement algorithms work towards minimizing possible risks brought by cloud architecture. They achieve this by, for example, separating VMs that belong to adversary users.

Most of the currently implemented VM placement algorithms are categorized in at least one of the three discussed categories. Very few stretch across two or more of the discussed categories. This results in secure cloud environments with poor service delivery to paying customers. In other situations, placement of VMs might provide high QoS at reduced operational costs, but with very low security. This study, therefore proposes a security-aware VM placement algorithm that also takes into consideration objectives that focus on cost-reduction and/or QoS.

4.4 Conclusion

This study proposes implementation of VM placement algorithm that can be classified into all discussed VM placement algorithms categories. This proposed VM placement algorithm, referred to as *O-sec VM placement algorithm* must consider security as its main objective because it is the theme of this study. In addition, it needs to consider other objectives classified under VM placement algorithms that focus on good QoS and cost-reduction. The approach towards implementation of *O-sec VM placement algorithm* is to augment the already existing VM placement algorithm. The idea is to find the VM placement algorithm that assumes higher optimization and/or security objectives as compared to others. This will further be modified towards implementation of O-sec VM placement algorithm. In order to find this VM placement algorithm, the currently available VM placement algorithms are evaluated. The next chapter therefore discusses currently implemented VM placement algorithms that are selected for evaluation in the study at hand.

Chapter 5

Selecion of VM Placement Algorithms for Evaluation

The previous chapter discusses different categories of VM placement algorithms. It further shows the need for implementation of O-sec VM placement algorithm. This is defined as VM placement algorithm that combines security objectives with objectives of the other two categories of VM placement algorithms. To successfully implement this kind VM placement algorithm, this study evaluates currently available VM placement algorithms. This is done to find VM placement algorithm that qualifies to further be augmented towards the notion of O-sec VM placement algorithms. This chapter, therefore, focuses on selection of the currently implemented VM placement algorithms. The next section discusses the selection process of VM placement algorithms to be evaluated. It further discusses in detail each of the selected algorithms. The researcher in this study constructs activity diagrams as a means to interpret each of the selected VM placement algorithms.

It is important to note that this chapter is work published by researcher of this study in paper titled *Towards O-Sec VM Placement Algorithms* [31]. This evaluation therefore includes only those VM placement algorithms selected in that publication.

5.1 Selected VM Placement Algorithms

In order to find VM placement algorithms that are evaluated in this chapter, a thorough search was made in different websites and online databases through use of the internet. Only those research efforts published in commonly used platforms are selected. These platforms include ACM, IEEE, Springer, Elsevier, Tailor and Francis, and Science Direct. About thirty (30) publications that focus on VM placement algorithms are found in this search. In order to reduce this number, elimination process is used based on abstract readings. This eliminates VM placement algorithms that are irrelevant to this study. Furthermore, an in-depth study of the remaining publications is done to reduce the number of VM placement algorithms. It reduces the final selection of publications to ten (10). The next subsections discuss 10 VM placement algorithms in detail.

5.1.1 Resource-based First Fit Algorithm (RFFA)

This is a VM placement algorithm proposed by Kuo et al. [35] called Resource-based First Fit Algorithm (RFFA). It focuses on energy consumption minimization that results in lower datacentre operational costs. It achieves this by allocating VMs to the first PM with enough physical resource capacities to host a requested VM. The algorithm uses four resource requirements to find a suitable PM. These resource requirements are CPU, memory, disk and network. For a PM to qualify to host requested VM, capacities of the four physical resources should be equal or greater than resource requirements of the requested VM. There is also a set $\boldsymbol{P}$, which is the set of PMs that host at least one VM. These are considered by this algorithm as active PMs. Those that are not included in this set are in idle mode and are therefore switched off to minimize energy consumption. The RFFA which originates from work by Kuo et al. [35] is interpreted in this study using activity diagram shown in Figure 5.1.

To explain how RFFA works, the algorithm checks all PMs within the set $\boldsymbol{P}$ to find a suitable PM to host a requested VM. If such a PM is found in the set $\boldsymbol{P}$, then placement is made. If there is no suitable PM, a new PM is powered-on and then added to the set $\boldsymbol{P}$. The placement is then made into this newly powered-on PM.

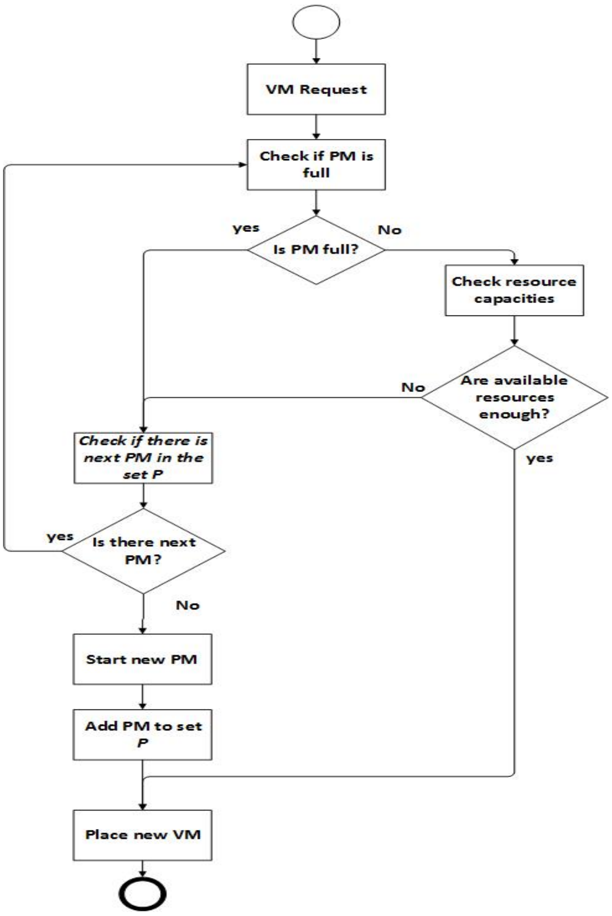


Figure 5.1: Resource-based First Fit Algorithm (RFFA)

5.1.2 VM Placement Algorithm to Minimize Physical Machine Count (VMP-MPMC)

This is a heuristic algorithm that is proposed by Ohta [48]. It focuses on energy consumption minimization while maintaining good QoS. This algorithm minimizes energy consumption by reducing number of powered-on PMs within the cloud infrastructure. It achieves this by using best fit placement to assign VMs to suitable PMs. That is, it chooses a PM with least available resources suitable to host requested VM. Furthermore, it also minimizes energy consumption by migrating VMs from under-utilized PMs and then switching them off. To provide QoS, this algorithm minimizes possible resource exhaustion by migrating VMs from over-utilized PMs. These over-utilization of PMs are brought by changing workloads within hosted VMs. Failure to sufficiently address resource exhaustion issue results in performance degradations. The algorithm is interpreted by researcher of this study using activity diagram shown in Figure 5.2.

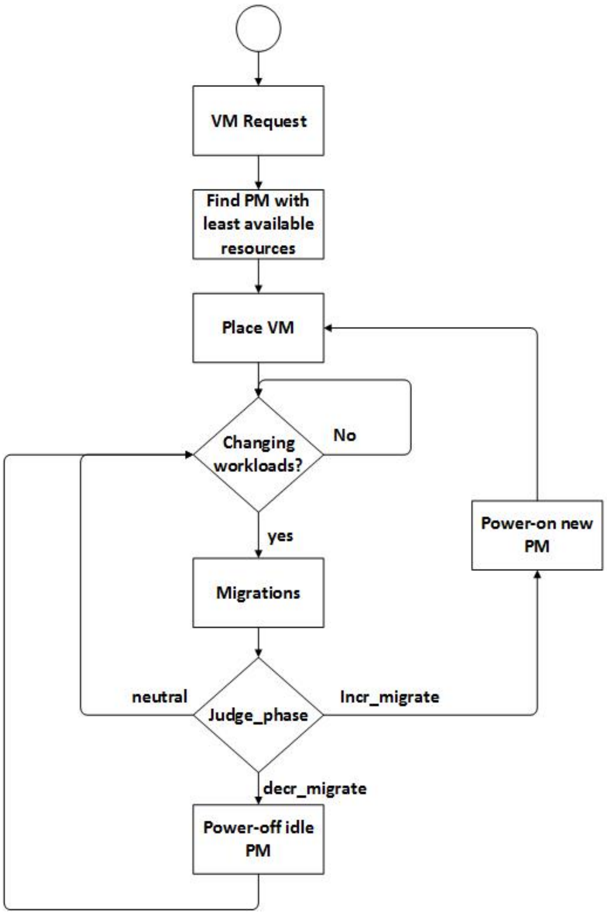


Figure 5.2: VM Placement Algorithm to Minimize Physical Machine Count

This algorithm shown in Figure 5.2 uses three procedures to successfully increase or decrease the number of powered-on PMs based on workload requirements. These procedures are: *judge_phase*, *incr_migrate* and *decr_migrate*.

- *judge_phase:* this is the procedure that determines whether dynamic reassignment requires change in the number of powered-on PMs. It calls one of the two other procedures depending on the need to either increase or decrease number of powered-on PMs.

- *incr_migrate:* this procedure increases number of powered-on PMs to allow placement of VMs.
- *decr_migrate:* this procedure decreases number of powered-on PMs in cases where migrations result in idle PMs.

The *judge_phase* procedure further uses *neutral* procedure if neither of the two (*incr_migrate* and *decr_migrate*) are required. This procedure means that migrations are made without the need to either increase or decrease number of powered-on PMs.

5.1.3 Dynamic Load Management (DLM) Algorithm

This is an algorithm proposed by Andreolini et al. [5] that focuses on migrations overhead reduction in order to provide good QoS. It is a load balancing VM placement algorithm that uses live migrations to evenly distribute VMs across the cloud infrastructure. However, these migrations are associated with overheads which devastate cloud architecture performance if excessively used. In order to minimize these migration overheads, the algorithm uses selective precisions to decide on which VMs are to be migrated. It achieves this by periodic monitoring of both load profiles of PMs and load trend behaviour of VMs. Migrations are therefore triggered when there is an intensive and persistent workload change within PMs. This algorithm is interpreted in this study by using activity diagram shown in Figure 5.3.

To successfully achieve its objectives, this algorithm uses the following four phases to migrate VMs:

- *Phase 1: Selection of sender hosts:* This phase uses the already discussed selective precision to determine fewer number of PMs with intensive and persistent workload change.
- *Phase 2: Selection of guests:* This phase selects VMs that need to be migrated from selected PMs determined in Phase 1. This selection of VMs uses three-step load trend-based model. In this model, the first step evaluates loads of each VM. The second step sorts VMs depending on workloads, and third step chooses subset of VMs with highest loads.

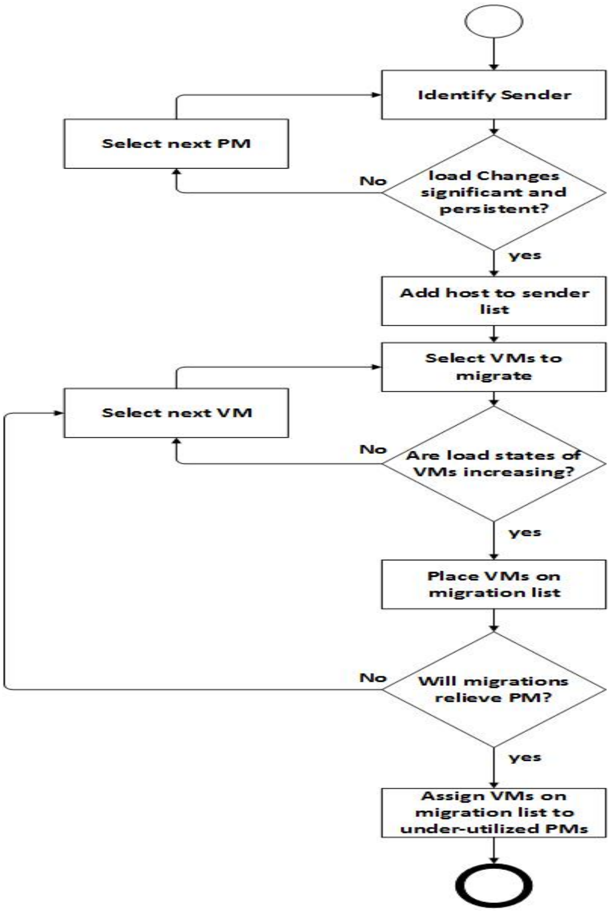

Identify Sender
Select next PM
No
load Changes significant and persistent?
yes
Add host to sender list
Select VMs to migrate
Select next VM
No
Are load states of VMs increasing?
yes
Place VMs on migration list
No
Will migrations relieve PM?
yes
Assign VMs on migration list to under-utilized PMs

Figure 5.3: Dynamic Load Management Algorithm

- *Phase 3: Selection of receiver hosts:* This phase chooses new PMs to host migrated VMs. The receiver PMs are chosen to be those that have most available resource capacities . To avoid overloading that might result in excessive migrations, each of the migrated VMs is assigned a different receiver PM.

- *Phase 4: Assignment of guests:* The selected VMs in Phase 2 are assigned to receiver PMs by using classical greedy method. The algorithm starts with VMs that have highest loads and assign them to PMs with most available resource capacities.

5.1.4 Traffic and Power-aware VM Placement Algorithm (TPVMP)

This VM placement algorithm proposed by Vu and Hwang [67] focuses on both cost reduction minimization and good QoS. It reduces energy consumption by consolidating VMs into *high capacity* PMs and switching off as many *low capacity* PMs as possible. These *high capacity* PMs are PMs that consume lower energy to process a unit workload as compared to *low capacity* PMs. To provide good QoS, the algorithm minimizes overall traffic cost within the cloud infrastructure. It achieves this by placing communicating VMs with high traffic in PMs that are in close proximity to each other. The algorithm further takes into consideration dynamic reassignment of VMs using migrations to minimize chances of under-utilization and over-utilization of PMs. The algorithm is analyzed in this study by using activity diagram shown in Figure 5.4. The *high capacity* and *low capacity* PMs are referred to as HC and LC PMs respectively in the figure.

This algorithm proposed by Vu and Hwang [67] is further explained in this study by using the following simple steps.

- *Step 1:* for allocation of VM, this algorithm first checks if there are any active PMs suitable to host this VM. If there are no suitable PMs, then the algorithm selects one of the inactive PMs to host this newly requested VM. These inactive PMs are those that had no allocated VMs and therefore are switched off to minimize energy consumption.

- *Step 2:* for a PM selected as suitable to host the VM, the algorithm further checks if the allocation will result in over-utilization of the PM. If this is the case, then the different PM is selected, otherwise an inactive PM is powered on. It is important to note that *high capacity* PMs are given first priority in the selection process. The *low capacity* PMs are selected only when there are no *high capacity* PMs suitable to host VM.

- *Step 3:* the algorithm then checks if the VM to be placed has any sibling VMs allocated to any of the active PMs. These sibling VMs are those that communicate with VM to be allocated. If there are no sibling VMs, the allocation to selected PM is completed. But, if there are sibling VMs, then the algorithm checks if distance between selected PM and PMs hosting siblings exceeds maximum required distance.

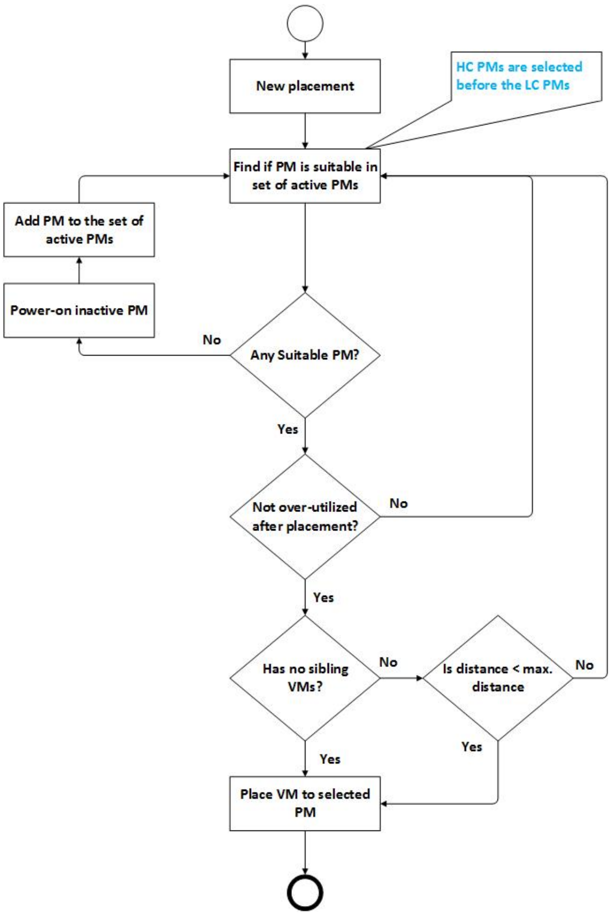

New placement
HC PMs are selected before the LC PMs
Find if PM is suitable in set of active PMs
Add PM to the set of active PMs
Power-on inactive PM
No
Any Suitable PM?
Yes
Not over-utilized after placement?
No
Yes
Has no sibling VMs?
No
Is distance < max. distance
No
Yes
Yes
Place VM to selected PM

Figure 5.4: Traffic and Power-aware VM Placement (TPVMP) algorithm

That being the case, then a different PM is selected and the process starts over again.

Note: This allocation steps work for both newly requested VMs and VMs selected for migrations.

5.1.5 Migration Based VM Placement (MBVMP) Algorithm

MBVMP is a VM placement algorithm proposed by Li et al. [37]. It focuses on providing good QoS by minimizing total job completion time of placing a newly requested VMs. According to authors who proposed this algorithm, a VM takes different completion times to be placed in different PMs which have different resource capacities. So they propose the VM placement algorithm that selects PMs which take least completion time to place a newly requested VM. This proposed algorithm further uses migrations if there are no enough resources in the selected PM to host newly requested VM. There is however, the constraint that governs placement of VMs using migrations in MBVMP. The constraint states:

"*MBVMP considers migrations only if the total completion time, which includes migration time, is less than the time taken for direct placement to the available PM with enough resources to host the newly requested VM*"

The researcher of this study uses activity diagram shown in Figure 5.5 to interpret MBVMP algorithm.

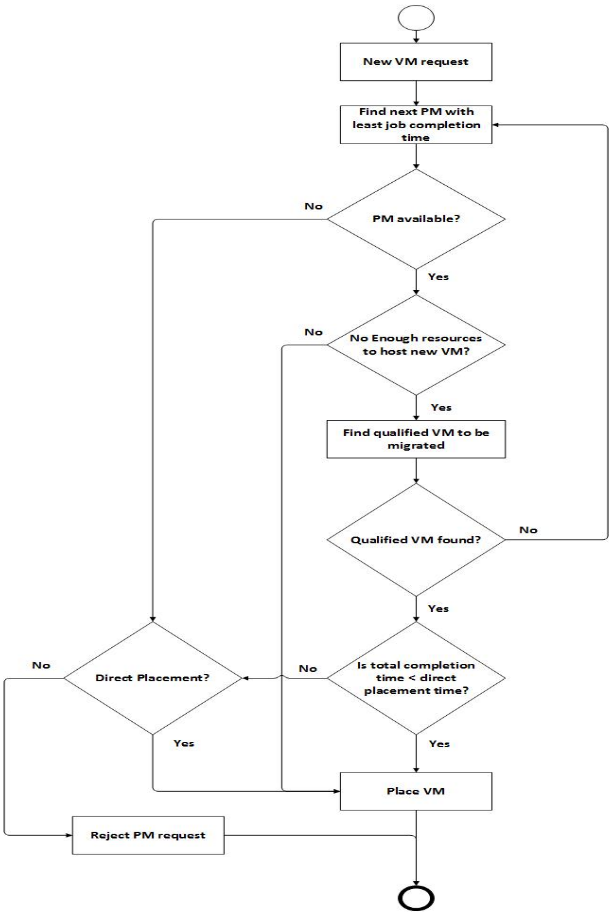

New VM request
Find next PM with least job completion time
PM available?
No
Yes
No Enough resources to host new VM?
No
Yes
Find qualified VM to be migrated
Qualified VM found?
No
Yes
Is total completion time < direct placement time?
No
Yes
Direct Placement?
No
Yes
Place VM
Reject PM request

Figure 5.5: Migration Based VM Placement (MBVMP) algorithm

In order to explain MBVMP algorithm shown in Figure 5.5, this study uses the following steps:

- *Step 1:* The algorithm sorts PMs in order of their job completion times in placement of VMs.
- *Step 2:* For new VM request, the algorithm checks if there are enough resources within the first PM that takes least completion time to host VM. If there are enough resources, the placement is completed.
- *Step 3:* In cases where there are no enough resources in the first selected PM, the algorithm searches for VM that qualifies to be migrated. This VM qualifies based on the three things: 1) there must be enough resources released after migration , 2) there must be a PM with enough resources to host this migrated VM, and 3) total job completion time which includes time to migrate VM must be less than that of direct placement into other available PMs. If this VM is found, then placement using MBVMP is completed. However, if direct placement takes less completion time compared to MBVMP then, placement is completed using direct placement.
- *Step 4:* If there are no qualified VMs to be migrated in selected PM, the algorithm iterates through the list of sorted PMs to find one with qualified VMs. If none of the PMs contain qualified VMs and there are no possibilities of direct placements, request is rejected.

5.1.6 Security-aware VM Placement using Metrics (SVMP-M)

Caron and Toinard [15] propose a VM placement algorithm that focuses on secure placement of VMs. It takes user security requirements as part of its input to define security level needs of cloud users. These security requirements are presented as *bitsets* vector metrics which are interpreted by the scheduler. Each *bit* in the vector is set to either "1" if the security characteristic is required or otherwise "0". To successfully use the metrics, CSPs need to provide security capacities of their cloud infrastructure by use of vector metrics. For example, the security processes vector for PMs can be represented as: vectorProcess = [Anti-virus IDS Firewall]. With known security capacities of the cloud infrastructure, cloud users detail their security requirement which are presented using *bitset* vectors. The user security requirement for the given example can be presented as: vectorProcess = [1 0 1], which shows the need for an anti-virus and firewall.

To explain how this algorithm by Caron and Toinard [15] works, the researcher in this study constructs activity diagram shown in Figure 5.6.

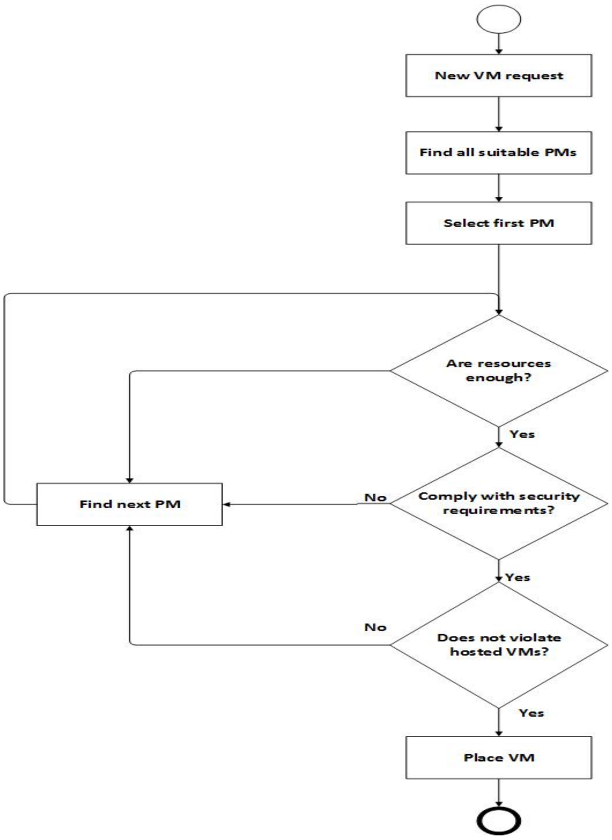


Figure 5.6: Security-aware VM Placement using Metrics

This activity diagram, shown in Figure 5.6 is further explained using the following steps:

- *Step 1:* For new VM request, the algorithm identifies all PMs that are compatible with the newly requested VM. For PM to be compatible, it needs to: 1) have enough resources to accommodate the newly requested VM, 2) have security characteristics that comply with user security requirements provided, and 3) have hosted VMs that will not be violated by placement of newly requested VM.

- *Step 2:* The algorithm further selects the first PM with enough resources from list of PMs identified in Step 1.

- *Step 3:* The algorithm places newly requested PM then updates infrastructure configurations to include newly placed VM.

5.1.7 Modified k-means Clustering VM Placement (MKCVMP) Algorithm

This proposed VM placement algorithm by Chowdhury et al. [16] focuses on both providing good QoS and reducing running costs. It achieves good QoS by migrating VMs from over-utilized PMs to avoid resource exhaustions. It further migrates VMs from under-utilized PMs and switches them off to reduce running costs. This algorithm achieves these by exploring clustering technique. It clusters VMs in migration lists based on their CPU utilization and RAM allocation. The VMs are then placed into available PMs, starting with those that are in most dense cluster. The proposed algorithm is interpreted in this study by using activity diagram shown in Figure 5.7.

This algorithm interpreted using Figure 5.7 uses 3-step approach in: identifying over-utilized or under-utilized PMs, selection of VMs to be migrated, and placement of VMs into new PMs. These steps are:

- *Step 1:* This step of the algorithm is the identification of PMs that are either over-utilized or under-utilized. The algorithm uses thresholds to identify these PMs.

- *Step 2:* This is the step that selects VMs to be migrated away from under-utilized and over-utilized PMs identified in step 1. The algorithm uses Minimum Migration Time (MMT) policy as one of its selection methods. This is the policy that selects VMs that take minimum amount of time to be migrated. The selected VMs are added to the migration list.

- *Step 3:* This is the step that allocates VMs in the migration list to available PMs. Unlike most energy-aware VM placement, this algorithms **does not** use power-aware best-fit VM placement to successfully place VMs into available PMs.

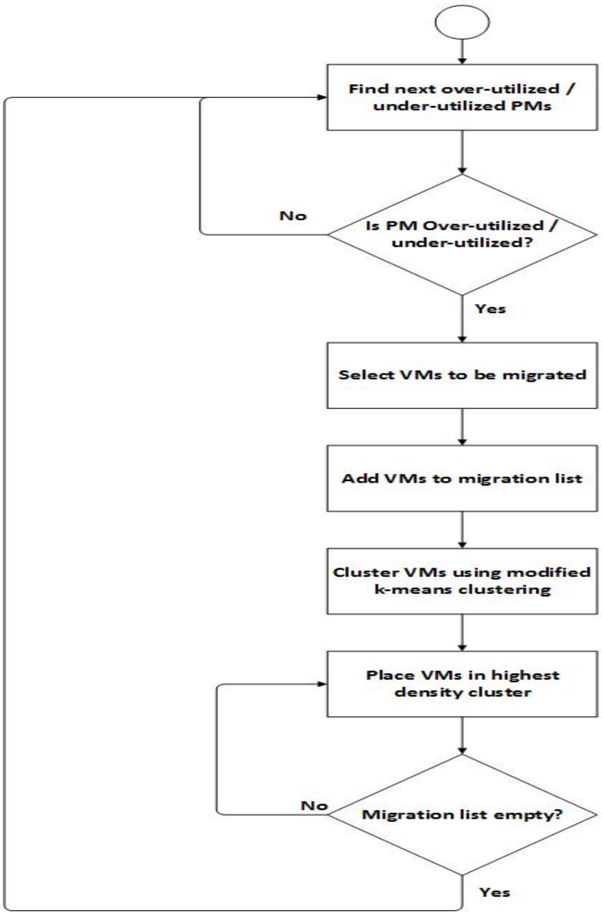

Find next over-utilized / under-utilized PMs
Is PM Over-utilized / under-utilized?
No
Yes
Select VMs to be migrated
Add VMs to migration list
Cluster VMs using modified k-means clustering
Place VMs in highest density cluster
Migration list empty?
No
Yes

Figure 5.7: Modified k-means Clustering VM Placement Algorithm

That is, it does not place VMs into PMs that use lower energy to consume unit workload. Instead, this proposed algorithm modifies this step by using modified k-means clustering technique. It clusters VMs in migration list based on their CPU utilization and RAM allocation. After clustering, the modified bin packing algorithms are used to allocate VMs to available PMs. Those VMs in highest density clusters are placed first.

5.1.8 Min Cut Ratio-aware VM Placement (MCRVMP) Algorithm

This VM placement algorithm that is proposed by Biran et al. [9] focuses on good QoS. It uses the concept of connected components to minimize possibilities of traffic bursts within cloud datacentres. To achieve this, the algorithm consolidates communicating VMs into PMs that are in close proximity to each other. This minimizes total traffic cost within the cloud infrastructure. This algorithm is used as an example of VM placement algorithms that focus on QoS in Chapter 4. Therefore, detailed explanation of this algorithm is in that chapter. It also includes interpretation of the algorithm using activity diagram, which is shown in Figure 4.1.

5.1.9 Sep-pack and Dynamic Algorithm

This is a VM placement algorithm proposed by Yang et al. [75]. It is modelled as a modified bin-packing algorithm that strives towards minimizing datacentre operational costs while maintaining good QoS. In this algorithm, PMs are taken to be bins while VMs are items. Items are further categorized into green items that represent CPU-intensive VMs and red items that represent data-intensive items. The algorithm consolidates both green and red items into a fewer number of bins in order to minimize energy consumption. Whenever there are needs for migrations, this algorithm migrates only CPU-intensive VMs. This is because migrating data-intensive VMs away from their images degrades their performances up to 40 percent. But migrations of CPU-intensive VMs away from their images have insignificant impact to their performance.

Sep-pack and dynamic algorithm [75] is used as an example of algorithms that focus on cost reduction discussed in Chapter 4. The detailed explanation of this algorithm is therefore provided in that chapter. There is also an activity diagram that interprets the algorithm, and is shown in Figure 4.2.

5.1.10 A Security-aware Scheduler (SS)

This is the VM placement algorithm that is proposed by Afoulki et al. [2]. It focuses on secure placement of VMs within the cloud infrastructure. This algorithm takes as part of its input, adversary user lists submitted by users as part of their requirements. These lists are used to generate incompatibility groups used by VM scheduler to isolate VMs belonging to adversary users. The algorithm is used in Chapter 4 as an example of VM placement algorithms that focus on security. It is therefore explained in detail in that chapter. There is also an activity diagram used in this study to interpret the algorithm, which is shown in Figure 4.3.

5.2 Summarizing the 10 VM placement algorithms

This section provides the summary of the 10 selected VM placement algorithms. The summary is presented in tabular format as shown in Table 5.1.

5.3 Conclusion

Having selected the VM placement algorithms, the next step is the evaluation process. This is required to find the VM placement algorithm that will further be augmented towards the notion of Optimized security-aware VM placement algorithm. This study uses quantitative evaluation criteria to find suitable VM placement algorithm that will further be augmented. This criteria is discussed in detail in the next chapter.

Table 5.1: Summary of the selected VM placement algorithms

Names	**Categories**	**Statement**
RFFA	cost reduction	It allocates VMs to the first available PM with enough resources in order to minimize machine count.
VMP-MPMC	cost reduction good QoS	It uses best-fit placement to minimize number of powered-on PMs. It also reduces physical resource exhaustion by migrating VMs from over-utilized PMs.
DLM	good QoS	It uses live migrations to evenly distribute VMs across the infrastructure. It considers selective precision to minimize migration overheads.
MCRVMP	good QoS	Uses connected components to minimize possibilities of traffic bursts.
SEP-Pack	cost reduction good QoS	Consolidates VMs into fewer PMs to reduce energy consumption. It restricts migration of data-intensive VMs away from their images to minimize performance degradations.
TPVMP	cost reduction good QoS	It reduces energy consumption by consolidating VMs into high capacity VMs. It minimizes overall traffic cost by placing communicating VMs in close proximity to each other.
MBVMP	good QoS	It considers PMs with least job completion time for placement of VMs.
SS	security	It separates VMs belonging to adversary users.
MKCVMP	cost reduction good QoS	It migrates VMs from under-utilized PMs and switch off idle PMs. It Migrates VMs from over-utilized PMs to avoid resource exhaustion
SVMP-M	security	It takes user security requirements as input to define the required security levels

Chapter 6

Evaluation of the Currently Implemented VM Placement Algorithms

The previous chapter (Chapter 5) discusses selection of the currently available VM placement algorithms. It further discusses in detail the reflected optimization and security objectives that are reflected in each of these VM placement algorithms. In this chapter, the selected VM placement algorithms are evaluated to find the best that can be improved towards the notion of O-Sec VM placement algorithm. The evaluation criteria used in this study is a quantitative evaluation. It rates availability of optimization and/or security objectives reflected in the selected VM placement algorithms. This criteria uses a revised process introduced in the article by Eloff [18]. It uses a formula to calculate percentage score of reflected objectives in each of the selected VM placement algorithms. In order to successfully use the formula, this evaluation criteria needs to involve identification of objectives observed in each VM placement algorithm. It further requires scaling of these identified objectives based on their importance.

6.1 Identification of Objectives

The first step towards selection of VM placement algorithm is to identify important objectives that a qualified VM algorithm needs to observe. These objectives are selected from a pool of optimization and security objectives based on their importance. They are selected based on their contribution towards cloud computing characteristics defined in this study. In addition, they are also selected based on their contribution towards implementation of VM placement algorithms classified under three discussed categories. The identified objectives are: 1) elasticity, 2) response time, 3) energy consumption minimization, 4) network traffic minimization, 5) reduced SLA violations and 6) security.

6.1.1 Elasticity or Dynamic Reallocation

Elasticity, which is also referred to as dynamic reallocation or dynamicity, is the automated reallocation of VMs within the cloud infrastructure [11]. It is considered in this study as the *critically important* objective. This is because it makes it possible for *self-managing* characteristic to be implemented in the cloud. As mentioned in previous chapters, this characteristic is normally used to avoid unwanted circumstances that jeopardize the optimization and/or security objectives observed in the clouds. These unwanted circumstances include possibilities of over-utilization or under-utilization of resources brought by changing workloads [5]. They also include possibilities of undesired co-locations brought by aggregation of VMs [76]. Based on these, it is possible to use this objective to implement VM placement algorithms that are classified in any of the three discussed VM placement algorithm categories: VM placement algorithms that focus on cost reduction, VM placement algorithms that focus on QoS, and VM placement algorithms that focus on security. Using examples, it is possible to reduce datacentre running costs by minimizing under-utilization of resources and therefore using fewer machine counts. Also, it is possible to minimize over-utilization of resources hence providing good QoS. Lastly, it is possible to minimize undesired co-location by implementing security-aware reallocations.

6.1.2 Response Time

By virtue of having these two characteristics: *on-demand self service* and *utility*, cloud computing requires response time as an important tool to satisfy paying customers. Like all computing environments, delays are accepted only up to a certain level. The unexpected and persistent delays frustrate end users [55]. Also, they result in violations of expected performance levels that are normally stipulated in SLA [4]. One of the factors that cause these delays is an uneven placement of VMs within the cloud infrastructure. For example, unmanaged placement of communicating VMs within the cloud infrastructure results in high traffic costs that affect response time [44]. Although not critical, it is *important* to consider response time in the implementation of VM placement algorithms. This is to avoid SLA violations while satisfying paying customers.

6.1.3 Energy Consumption Minimization

Energy consumption is a world-wide concern due to the rapidly increasing sizes of datacentres [75]. Some studies estimate that approximately 70 percent of the total datacentre management costs are energy consumption costs [7]. This shows that energy consumption is the contributing factor to the daily running expenses of datacentres. It is therefore *critically important* to consider ways of reducing energy consumptions within cloud datacentres.

6.1.4 Network Traffic Minimization

It is mentioned in previous chapters that one of *shared resources* in cloud computing are internal networks. Like all resources, over-utilizing these internal networks result in compromise to QoS. High internal traffic between communicating VMs normally results in traffic bursts, more especially in peak times [9]. This causes unnecessary downtimes of hosted applications, therefore affecting expected QoS. It is therefore *critically important* to consider internal traffic minimization in implementation of VM placement algorithms [67].

6.1.5 Reduced SLA violations

In order to ensure good QoS, there is an SLA contract signed between cloud users and CSPs. This document entails expected minimum performance levels and mitigation actions in cases of unwanted incidences [4]. In addition, it also stipulates consequences in cases of breach of this SLA. These consequences are normally in the form of penalties whereby CSPs pay for breach of the contract [23]. To minimize these unwanted costs, CSPs need to ensure that placement of VMs does not result in violations of this SLA.

6.1.6 Security

This objective is the main theme of this study. It focuses on minimizing risks that are brought by clouds' architectural vulnerabilities [23]. These vulnerabilities include, among others, undesired co-location of VM brought by *aggregate* characteristic of cloud computing [2]. Being the main theme of this study makes security objective most important of the identified objectives. However, the objective is rated lowest in evaluations of selected VM placement algorithms. This is to minimize chances of selecting VM placement algorithm that observes security as one of its objectives. The aim is to enforce security as part of modifying a selected VM placement algorithm.

Having discussed the identified optimization and security objectives in this section, the next step is to rate these objectives based on their importance.

6.2 Rating of the Identified Objectives

This section performs rating of identified objectives based on their importances. This is done using constant values from 1 - 4 to represent different level of importance, as shown below.

$4 = critical\ importance,$

$3 = very\ important,$

$2 = important, and$

$1 = necessary$

The ratings of objectives based on their importance is shown in tabular format using

Table 6.1.

Table 6.1: Objectives Rating Table

Objectives	**Ratings**
Elasticity	4
Response time	2
Energy Consumption Minimization	4
Network Traffic Minimization	4
Reduced SLA Violations	3
Security	1

6.3 Scaling of Reflected Objectives

This section identifies reflected objectives in each of the selected VM placement algorithms. It further scores these identified objectives based on their existence in the algorithm. The scale shown below is used to show existence of the objectives in each of the 10 VM placement algorithms.

$2 = objective\ exists\ fully,$

$1 = objective\ partially\ exist, and$

$0 = does\ not\ exist$

Table 6.2 shows reflected objectives and their existence for each selected VM placement algorithm.

6.4 Evaluating VM Placement Algorithms for Final Selection

As mentioned, final selection process uses formula to calculate percentage score of reflected objectives for all selected VM placement algorithms. It uses combination of rat-

Table 6.2: Objectives Evaluation Table

VM Placement Algorithm	**Objective Evaluation**	
Resource-based First-Fit Algorithm (RFFA)	Elasticity	2
	Response Time	0
	Energy Consumption	2
	SLA Violations	0
	Network Traffic	0
	Security	0
VM placement algorithms to minimize PM count (VMP-MPMC)	Elasticity	2
	Response Time	2
	Energy Consumption	2
	SLA Violations	0
	Network Traffic	0
	Security	0
A Security-aware Scheduler (SS)	Elasticity	2
	Response Time	0
	Energy Consumption	0
	SLA Violations	2
	Network Traffic	0
	Security	2
Security-aware VM Placement using Metrics (SVMP-M)	Elasticity	0
	Response Time	0
	Energy Consumption	0
	SLA Violations	2
	Network Traffic	0
	Security	2
Dynamic Load Management (DLM)	Elasticity	2
	Response Time	0
	Energy Consumption	0
	SLA Violations	0
	Network Traffic	0
	Security	0
Min Cut Ratio-aware VM Placement (MCRVMP)	Elasticity	0
	Response Time	0
	Energy Consumption	0
	SLA Violations	0
	Network Traffic	2
	Security	0
SEP-Pack and Dynamic Algorithm	Elasticity	2
	Response Time	1
	Energy Consumption	2
	SLA Violations	0
	Network Traffic	0
	Security	0
Traffic and Power Aware VM Placement (TPVMP)	Elasticity	2
	Response Time	0
	Energy Consumption	2
	SLA Violations	0
	Network Traffic	2
	Security	0
Migration Based VM Placement (MBVMP) Algorithm	Elasticity	2
	Response Time	2
	Energy Consumption	0
	SLA Violations	0
	Network Traffic	0
	Security	0
Modified k-means Clustering in VM Placement (MKCVMP)	Elasticity	2
	Response Time	0
	Energy Consumption	1
	SLA Violations	0
	Network Traffic	0
	Security	0

ings of the objectives shown in Table 6.1 and reflected objective scores shown in Table 6.2. The next subsection shows how this formula is derived.

6.4.1 Deriving the Formula

To calculate percentage score of the reflected optimization and security objectives, the following equation is used:

$$O_j = \frac{ObjectiveEvaluation}{TotalObjectiveRating} \times 100 \tag{6.1}$$

$$O_j = \frac{\sum_{i=1}^{N} x_i k_i}{\sum_{i=1}^{N} x_{i(TOT)} k_{i(TOT)}} \times 100 \tag{6.2}$$

Where,

x_i is the evaluation factor allocated to the presence of the objective,

k_i is the importance rating of each objective,

$x_{i(TOT)}$ is the evaluation factor allocated to the presence of the objective in an ideal situation,

$k_{i(TOT)}$ is the importance rating for an ideal situation, and

N is total number of objectives used for evaluation.

For an ideal VM placement, $x_{i(TOT)} = 2$ for all placement algorithms. This is brought by an assumption that objectives fully exist in ideal situations. From this, the formula becomes,

$$O_j = \frac{\sum_{i=1}^{N} x_i k_i}{2 \sum_{i=1}^{N} k_{i(TOT)}} \times 100 \tag{6.3}$$

And therefore,

$$O_j = 50 \frac{\sum_{i=1}^{N} x_i k_i}{\sum_{i=1}^{N} k_{i(TOT)}} \tag{6.4}$$

From ratings conducted in Section 6.2, the value of $k_{i(TOT)} = 18$ and $N = 6$, therefore,

$$O_j = 0.46 \sum_{i=1}^{N} x_i k_i \tag{6.5}$$

For all,

$$1 \leq j \leq M$$

Where M is total number of evaluated placement algorithms

6.4.2 Using the Formula

Using the derived formula and substituting the x_i and k_i with values obtained in Tables 6.1 and 6.2, percentage score for each VM placement algorithm is shown in Table 6.3.

Table 6.3: VM Placement Algorithms Evaluation Results

VM Placement Algorithm	Score
RFFA	7.36
VMP-MPMC	9.2
DLM	3.68
MCRVMP	3.68
SEP-Pack	8.28
TPVMP	**11.04**
MBVMP	5.52
SS	7.36
MKCVMP	5.52
SVMP-M	3.68

From the table, VM placement algorithm that scores highest percentage is *Traffic and Power-aware VM Placement (TPVMP) algorithm.*

6.5 Summary

To summarize this chapter, ten VM placement algorithms are selected for evaluation. This evaluation aims to find VM placement algorithm that assumes most important optimization and security objectives. It uses quantitative evaluation that takes into consideration reflected objectives in each one of the selected VM placement algorithms. To achieve this evaluation, the criteria firstly identifies important objectives that are appropriate for implementation of O-Sec VM placement algorithm. Next, it rates these objectives based on their importance. This rating uses scale between 1 and 4, where 1 shows that the objective is necessary and 4 shows that the objective is critically important. In the third step, it identifies and scores objectives based on their existence using scale of 0 - 2 (0 = does not exist, 1 = partially exist, and 2 = fully exist). The last step uses the formula to evaluate VM placement algorithms based on reflected objectives.

6.6 Conclusion

The evaluations conducted in this chapter find TPVMP as the VM placement algorithm that qualifies to further be augmented towards implementation of O-sec VM placement algorithm. This is the VM placement algorithm that observes network traffic minimization and energy consumption minimization as its objectives. For better understanding of this VM placement algorithm, the next chapter discusses it in detail. It further discusses how the TPVMP algorithm is augmented to implement O-sec VM placement algorithm.

Chapter 7

TPVMP as a Candidate to be considered for O-Sec VM Placement

The previous chapter evaluates ten currently available VM placement algorithms. These are evaluated using a quantitative approach to find a VM placement algorithm that assumes highest optimization and/or security objectives. The aim is to augment the selected VM placement algorithm towards the notion of *O-Sec VM placement*. According to the evaluation criteria, TPVMP qualifies to be a VM placement algorithm that showed potential to be further augmented for security purposes. The current chapter discusses this VM placement algorithm in detail, reflecting on the objectives observed in its implementation. It further elaborates on how these objectives are augmented to achieve the goals of this study.

7.1 TPVMP - from Researcher's Perspective

There are two main objectives considered in the implementation of TPVMP. These objectives, which are both optimization objectives, are: energy consumption minimization and network traffic minimization. Energy consumption minimization on one hand, strives to reduce total operational costs of cloud datacentres. It achieves this by reducing number of powered-on PMs in order to minimize energy consumption. Network traffic minimization on the other hand, strives to provide good QoS by minimizing network traffic that normally results in unwanted latencies. To successfully achieve these main objectives, TPVMP also considers elasticity as its third objective. It uses migrations for dynamic reallocation of VMs in order to minimize both energy consumption rates and network traffic. Based on these observed objectives, TPVMP is categorized under two types of the discussed VM placement algorithms groups. These are VM placement algorithms that focus on good QoS, and VM placement algorithms that focus on cost reduction.

In order to understand TPVMP objectives and how they are implemented, the next subsections discuss them in detail. These discussions include possible vulnerabilities associated with these objectives. Also, they include countermeasures that overcome risks brought by the stated vulnerabilities.

7.1.1 Energy Consumption Minimization

As mentioned, one of the main objectives considered in implementation of TPVMP is energy consumption minimization. This objective consolidates VMs into fewer number of PMs in order to reduce machine count. The consolidation is, however, different from the commonly used method that packs VMs into first available PMs. In this method, VMs are consolidated into *high capacity* PMs as much as possible. The *low capacity* PMs are considered for placement only when high capacity PMs have limited resources to host VM requests. To explain these two types of PMs:

- *High capacity PMs*: These are PMs with higher CPU usage per energy consumption. This means that high capacity PMs use low energy consumption to process a unit workload.

- *Low capacity PMs*: These are PMs with lower CPU usage per energy consumption. It means that low capacity PMs use higher energy consumption to process a unit workload as compared their counter partners.

From these explanations, it means consolidating VMs into high capacity PMs reduces overall energy consumption within cloud datacentre. In order to maintain these low energy consumptions, TPVMP also considers dynamic reallocations of VMs known as *elasticity*. Through this, VMs are migrated from low capacity into high capacity PMs as much as possible.

Vulnerabilities associated with Energy Consumption Minimization Objective

There are possible vulnerabilities identified in this study which are associated with this objective. These vulnerabilities are brought by consolidation of VMs into fewer number of PMs. They include the following:

- **Unwanted co-location of VMs**: As discussed, *aggregate* characteristic makes undesired co-locations of VMs possible. These include co-location of VMs with different vulnerability statuses as explained by Yuchi et al [76]. In such cases, VMs with high vulnerability statuses compromise security of those with low vulnerability statuses. Malicious users take advantage of highly vulnerable VMs and use shared resources to attack less vulnerable co-located VMs.

- **Possible co-location of adversary users**: There are also possibilities of co-locations of adversary users in cloud environments. Once co-located, it is possible for adversary users to take advantage of shared resources to compromise either confidentiality, integrity or availability of VMs belonging to other users [25].

Security Features to augment Energy Consumption Minimization Objective

The possible security features identified to augment vulnerabilities brought by energy consumption minimization include the following:

- The consolidation of VMs into fewer PMs may include adversary user constraint which states that "*VMs belonging to adversary users should not reside within the same PMs*".

- The vulnerability status of VMs may be used in their placement, as proposed by Yuchi et al. [76]. According to the last cited article, VMs are separated based on their vulnerability statuses. The more vulnerable VMs are separated from the less vulnerable to avoid possible compromise of secure VMs. This idea is based on the statement that "*The security level of the PM is equal to the security level of the weakest VM hosted in that PM*".

7.1.2 Network Traffic Minimization

The second objective considered in implementation of TPVMP is network traffic minimization. This objective reduces network traffic costs by placing communicating VMs into PMs that are in close proximity to each other. Examples of these communicating VMs include components of three-tier applications. These components need to be placed closer to each other in the cloud infrastructure. The idea is to minimize total traffic costs within the cloud infrastructure which normally result in communication delays. Failure to sufficiently reduce this traffic costs normally affects applications throughput. This results in compromise to the expected QoS as stipulated in the SLA. According to authors of TPVMP, total traffic cost within the cloud infrastructure is represented as:

$\sum_i \sum_j W_{ij} C_{ij} T.$

Where,

W_{ij} is the traffic weight between two VMs, VM_i and VM_j.

C_{ij} is the communication cost between two VMs, VM_i and VM_j.

T is time.

In this equation, communication cost C_{ij} is the only variable that can be changed to reduce total traffic cost. This is only possible through dynamic reallocation (elasticity) of these VMs. Placing two communicating VMs closer to each other reduces communication cost C_{ij}. This means, placement of communicating VMs into PMs that are in close proximity reduces total traffic cost within the cloud infrastructure. They must be placed at least a distance x away from each other. This distance x is normally measured using number of *hop counts* (number of network devices between communicating components). It is, however impossible to place all communicating VMs within same partition of the network. Therefore, authors of TPVMP prioritize placement of communicating VMs by starting with the heavily communicating VMs.

Vulnerabilities associated with Network Traffic Minimization

The possible vulnerabilities identified which are associated with this objective include the following:

- **Compromise of VM security levels**: Placement of VMs into PMs which are in close proximity to each other do not take into consideration security levels of those PMs. This means, reallocation of VMs based on the discussed objective might result in migration of critical VMs from secure to unsecure PMs. Migrating these critical VMs into vulnerable PMs compromises security of those VMs. In addition, it also compromises security of their communicating counter partners.

- **Undesired co-locations and possible adversary user co-locations**: As it is the case with the first objective, placement of VMs using network traffic minimization does not take into consideration possible co-location of adversary users. In addition, it does not include separation of VMs to avoid undesired co-locations.

Security Features to augment Network Traffic Minimization Objective

The possible security features that can expand this objective include the following:

- The placement of VMs into PMs that are in close proximity to each other may include the constraint that "*VMs belonging to adversary users should not reside within same PMs*". This constraint requires submission of adversary user lists as part of user requirements, as proposed by Afoulki et al. [2]. These adversary user lists are used to create incompatibility lists that are used in placement of VMs.

- Also, placement into PMs that are in close proximity to each other may involve selection of PMs based on required user security levels. This idea proposed by Caron et al. [15] requires submission of user security requirements. These include, among others, choice of operating systems, selection of anti-viruses and IDSs.

7.1.3 Elasticity or Dynamic Reallocation

Although it is not the main objective in the implementation of TPVMP, elasticity plays a vital role in supporting the two main objectives. The main role of this objective is to migrate VMs in order to achieve both energy consumption minimization and network traffic minimization objectives. As elasticity is not the main objective of TPVMP, possible vulnerabilities associated with it and security features that can be used to augment it are beyond the scope of this study.

7.2 Implementation of TPVMP

This chapter uses pseudo-code to interpret implementation of TPVMP because it is readable compared to previously used activity diagrams. Furthermore, it translates easily to generate actual coding of the algorithm. In order to demonstrate in-depth understanding of TPVMP, the researcher in this study uses own syntax to interpret it. The pseudo-code for TPVMP, as interpreted by the author of this study, is shown as Algorithm 1. It is important to note that this pseudo-code concentrates on selection of suitable PM to host VM request. This VM request includes both the newly requested VM (initial placement) and placement through migrations (dynamic allocation).

The author of this study uses readable English to illustrate TPVMP algorithm as shown in Algorithm 1. This is to ensure that the algorithm is easily interpreted without further explanations. However, there are keywords used in the pseudo-code that need explanations. These keywords are the following:

- *VM request*: This is an instruction to place a VM into the cloud infrastructure.
- *Available PMs*: This is a list of all powered-on PMs that host at least one VM. This list does not include idle PMs that are powered-off to minimize energy consumption.
- *Suitable PM*: This is a PM with enough resources to host a VM request.
- *New PM*: This is a PM that is powered-on and added to the Available PMs list. This happens when there are no suitable PMs in available PMs list to host VM request.
- *SiblingVMs*: These are the already placed VMs that are connected and communicate with the VM to be placed.
- *Allocated PM*: This is a PM that accepts and is allocated the VM request.

This pseudo-code is used throughout this study to interpret TPVMP. The augmentation towards O-Sec VM placement algorithm is made from the same algorithm.

Algorithm 1 TPVMP Algorithm

Input:

VM request; Available PMs;

Output:

Allocated PMs;

Description:

```
Order Available PMs based on energy consumptions; (high capacity PMs first)
Find suitable PM in Available PMs for VM request;
IF no suitable PM in Available PMs;
    Power-on new PM;
    Add new PM to Available PMs;
    Allocate VM;
IF suitable PM is NOT over-utilized after allocation of VM request;
    IF VM has no siblingVMs;
        Allocate VM to suitable PM;
    ELSE IF VM has siblingVMs;
        FOR each siblingVM, find hosting PM;
            Find distance between hosting PM and suitable PM;
            IF distance is less than required minimum distance;
                Allocate VM;
            ELSE IF distance is more than required minimum distance;
                Find next PM in Available PMs;
                IF distance is less than required minimum distance;
                    Allocate VM;
            ELSE Allocate VM;
ELSE Find next PM in Available PMs;
return Allocated PM;
```

7.3 Augmenting TPVMP with Security Features (O-Sec VM Placement Algorithm)

Having discussed TPVMP in detail, the next step is implementation of security features towards *O-sec VM placement algorithm*. This section discusses in detail these security features and how they minimize risks brought by undesired co-locations. There are two security features added to TPVMP in order to achieve secure placement. These are, *1) isolation of VMs belonging to adversary users*, and *2) isolation of critical and non-critical VMs*.

7.3.1 Isolation of VMs belonging to adversary users

As mentioned earlier, one of the challenges brought by both aggregate and shared resources is the possibilities of co-location of VMs belonging to adversary users. These co-locations promote inter-VM attacks that take advantage of shared resources to either disrupt, corrupt or spy other VMs. One of the resolutions to this situations is the secure placement that isolates VMs belonging to adversary users. TPVMP, like most of the currently implemented VM placement algorithms, does not take isolation of adversary users into consideration. Instead, it promotes adversary user co-location by consolidating VMs into fewer PMs regardless of their ownership. Based on this factor, the study finds it important to include *isolation of VMs belonging to adversary users* as one of the security features. This security feature is achieved by checking for VMs belonging to adversary users whenever a suitable PM to host VM request is found. The adversary user lists are compiled from lists submitted as part of user requirements. TPVMP with additional security feature of isolating VMs belonging to adversary users, referred to as *O-sec 1 VM placement algorithm*, is interpreted using pseudo-code in Algorithm 2. In this algorithm, modifications to the original TPVMP algorithm are shown using asterisk (*).

Note: The changes made in Algorithm 2 provide capabilities of the algorithm to search for VMs belonging to adversary users in suitable PMs. The placement into a suitable PM is completed only if there are no VMs belonging to adversary users.

Algorithm 2 O-sec 1 VM placement algorithm

Input: VM request; Available PMs; Adversary user list;

Output: Allocated PMs;

Description:

```
Order Available PMs based on energy consumptions; (high capacity PMs first)
Find suitable PM in Available PMs for VM request;
IF no suitable PM in Available PMs;
    Power-on new PM;
    Add new PM to Available PMs;
    Allocate VM;
IF suitable PM is NOT over-utilized after allocation of VM request;
    IF VM has no siblingVMs;
        *IF suitable PM has no adversary users' VMs*;
            Allocate VM to suitable PM;
    ELSE IF VM has siblingVMs;
        FOR each siblingVM, find hosting PM;
            Find distance between hosting PM and suitable PM;
            IF distance is less than required minimum distance;
                *IF PM has no adversary users' VMs*;
                    Allocate VM;
            ELSE IF distance is more than required minimum distance;
                Find next suitable PM in Available PMs;
                IF distance is less than required minimum distance;
                    * IF suitable PM has no adversary users' VMs*;
                        Allocate VM;
            ELSE Allocate VM;
ELSE Find next suitablePM in Available PMs;
return Allocated PM;
```

7.3.2 Isolation of critical and non-critical VMs

Novice to this study is the isolation of VMs that comprises of different security access levels. This is achieved by categorizing VMs into two groups; *critical* and *non-critical VMs*. Critical VMs, on one hand, are considered as VMs that require high security. The integrity and confidentiality of data within these VMs is of essence and therefore need controlled access. Examples of critical VMs are back-end database components of applications which store sensitive information. They are normally accessed through secure channels such as configured and controlled APIs. The security of these access channels are, however, beyond the scope of this study. Non-critical VMs, on the other hand, are those VMs whose access is open to a number of end-users. These are, for example, front-end components of applications that are accessed through insecure channels such as web browsers. The security access control level of non-critical VMs is lower compared to those of their counter partners. Co-locating these two types of VMs introduces threats to critical VMs. Malicious users with access to non-critical VMs take advantage of shared resources to bypass secure access controls of critical VMs. There is therefore the identified need to isolate these critical VMs from non-critical VMs. However, this isolation should take into consideration optimization objectives observed in TPVMP. More specifically, it should ensure that communicating components of the application are in PMs that are in close proximity to each other. Failure to ensure this may result in network traffic delays that jeopardize QoS.

The isolation of critical from non-critical VMs is achieved through the following:

- **Identification of PMs used for placement of critical VMs:** There is a need to identify PMs that are suitable to strictly host critical VMs. They must be a fraction of total PMs available in every leave of the network topology. This allows placement of non-critical VMs into PMs that are in close proximity to the PMs that host their critical communicating components.

- **Placement into PMs selected for critical VMs placement:** For new VM request, the algorithm places critical VMs into the identified PMs. If there are no suitable PMs to host a newly requested critical VM, request is suspended/rejected.

TPVMP with additional security features of isolation of VMs belonging to adversary users together with *isolation of critical from non-critical VMs* is demonstrated using pseudo-code as shown in Algorithm 3. This algorithm combines two security features discussed and it is referred to as *O-sec VM placement algorithm*. The information used to categorize VMs into critical and non-critical VMs is submitted as part of user requirements.

Note: In Algorithm 3, additional changes made from Algorithm 2 are shown using *. These changes allow O-sec VM placement algorithm to select appropriate PMs for critical and non-critical VMs.

7.4 Summary

This chapter is summarized in tabular format as shown in Table 7.1. The table shows main objectives reflected in implementation of TPVMP. It further shows possible vulnerabilities identified for each objective together with their possible countermeasures.

Table 7.1: TPVMP main Objectives, their Vulnerabilities and Countermeasures

Main Objectives	**Possible Vulnerabilities**	**Countermeasures**
Energy Consumption Minimization	Unwanted co-location of VMs	Isolation of VMs based on their vulnerability statuses.
	Possible co-location of adversary users	Isolation of VMs belonging to adversary users.
Network Traffic Minimization	Compromise of VM security levels	Isolation of VMs based on their security levels.
	Unwanted co-locations and possible adversary user co-locations	Isolation of VMs belonging to adversary users. Isolation of VMs based on their vulnerability statuses.

Note: These main objectives are achieved through use of the third objective, known as elasticity.

Algorithm 3 O-sec VM placement algorithm

```
Input:   VM request; Available PMs; Adversary user list; VM type;
Output:   Allocated PMs;

Description:
*Identify Critical PMs to host Critical VMs*;
*IF new request is Critical VM*;
    *Order Critical PMs based on energy consumptions*;
    *Make Critical PMs to be Available PMs*;
*ELSE order Non-Critical PMs based on energy consumption*;
    *Make Non-Critical PMs to be Available PMs*;
Find suitable PM in Available PMs for VM request;
IF no suitable PM in Available PMs;
    Power-on new PM;
    Add new PM to Available PMs;
    Allocate VM;
IF suitable PM is NOT over-utilized after allocation of VM request;
    IF VM has no siblingVMs;
        IF PM has no adversary users' VMs;
            Allocate VM to suitable PM;
    ELSE IF VM has siblingVMs;
        FOR each siblingVM, find hosting PM;
            Find distance between hosting PM and suitable PM;
            IF distance is less than required minimum distance;
                IF PM has no adversary users' VMs;
                    Allocate VM;
            ELSE IF distance is more than required minimum distance;
                Find next suitable PM in Available PMs;
                IF distance is less than required minimum distance;
                    IF suitable PM has no adversary users' VMs;
                        Allocate VM;
            ELSE Allocate VM;
ELSE Find next suitable PM in Available PMs;
return Allocated PM;
```

7.5 Conclusion

As shown in this chapter, the main objectives of TPVMP are energy consumption minimization and network traffic minimization. These are supported by the third objective, elasticity, which ensures that the main objectives are achieved. The study at hand identifies possible vulnerabilities brought by these objectives. Among them are those brought by undesired co-locations of VMs. These include undesired co-location of adversary users. Furthermore, they also include undesired co-location of VMs with different security levels. To counter for this, TPVMP is augmented with security features to implement the *O-sec VM placement algorithm*. The security features of choice are 1) those that isolate VMs belonging to adversary users, and 2) those that separate VMs with different security levels. Pseudo-codes on how to augment TPVMP are shown in this chapter. Having discussed these, the next step is implementation and evaluation of O-sec VM placement algorithm. There is, therefore a need to choose the platform for evaluation of the implemented algorithm. For this study, open-source cloud simulator called ***CloudSim Plus*** is chosen for evaluation process. To understand this simulator, the next chapter discusses it in detail.

Chapter 8

Developing A Simulated Cloud Environment for Experiments in this Research

This chapter provides an overview of the technical environment selected for evaluation of O-Sec VM placement algorithm. It discusses in detail the structure of the selected environment that makes it suitable for these evaluations. The environment of choice is a simulated cloud computing platform. This simulated platform is chosen because, experiments conducted on real infrastructure are costly, time consuming and sometimes limits scalability factor. Simulation environments, however, provide a faster way to run experiments that would take hours or days to run in real infrastructures [57]. For this study, CloudSim Plus simulation environment is selected as an appropriate platform for evaluation of O-Sec VM placement algorithm. The next section provides details on why CloudSim Plus is selected as an appropriate simulation environment. This is followed by discussions on generated datacentre architecture that is used for evaluations.

8.1 Selection of the Simulation Environment

There are a number of cloud simulation tools available for generation of cloud computing infrastructures. Most of these are used for evaluating performance of cloud datacentres based on placement of VMs. There are, however, some cloud simulation environments that have limitations with regard to extensibility and flexibility. These kind of cloud simulation environments make it impossible to be customized for evaluations conducted in this study. They include, among others, the following: Simulation Program for Elastic Cloud Infrastructure (SPECI), GreenCloud and Open Cloud Testbed (OCT).

- **Simulation Program for Elastic Cloud Infrastructure (SPECI)**: This simulation environment proposed by Sriram et al. [61] provides tests for scalability and performance aspects of cloud datacentres. It specifically evaluates VM placement algorithms that focus solely on QoS.

- **GreenCloud**: This simulation environment is proposed by Liu et al. [39]. It provides evaluations on energy consumption of datacentres brought by placement of VMs. It specifically concentrates on energy consumption rates that are produced by live migrations of VMs.

- **Open Cloud Testbed (OCT)**: This simulation environment is proposed by Grossman et al. [24]. It provides a simulated cloud environment with a fixed number of PMs. These PMs are spread across four datacentres that are in different geographical areas. The environment is used specifically for testing performance of distributed applications.

Different from the discussed cloud simulation environments is the most popular Java developed and open source CloudSim [13]. This is the cloud simulation environment that is widely used in most of the current studies that evaluate VM placement algorithms. It is a simulation environment of choice in evaluation of TPVMP proposed by Vu and Hwang [67]. CloudSim gained its popularity due to a number of its characteristics. These characteristics include flexibility and extensibility brought by open source packages that are programmed using a commonly known programming language. It allows its users to define additional classes in order to create scenarios suitable for their experimentation. However, CloudSim has a number of shortcomings that make it unfavorable for this study. These shortcomings include limited documentation, amount of duplicate code, absence of design patterns and lack of organized package structures [57]. All these shortcomings result in restricted extensibility and maintainability of the tool. They make it impossible to extend the tool without modifying its main classes. Due to these, the researcher of this study found it difficult to customize CloudSim to suit requirements of the evaluations.

In order to find cloud simulation environment suitable for this study, the researcher explored other cloud simulation environments currently available. Specifically, the researcher explored the extended versions of CloudSim. These are Cloud-Analyst, OpenSim and CloudSim Plus.

- **Cloud-Analyst**: This environment proposed by Wickremasinghe et al. [69] provides Graphical-User-Interface (GUI) version of CloudSim. It allows simulation of cloud datacentres that are in different geographical areas. It concentrates more on user traffic and response times of remotely located applications. It is not extensible to cover experiments required for this study.
- **OpenSim**: This tool proposed by Sitaram et al. [59] provides simulations specific to OpenStack cloud computing environment. Its main objective is to provide an overview of Openstack cloud and how it performs, without actual deployment on real infrastructure. This means OpenSim is not flexible and extensible as it must represent configurations and structures of Openstack cloud environment.
- **CloudSim Plus**: This is an extended version of CloudSim proposed by Silva Filho et al. [56] in 2017. It includes additional functionalities that provide a more user friendly simulation environment . It overcomes discussed shortcomings of CloudSim, making it a more extensible and flexible simulation environment. This is a cloud simulation environment of choice for this research and is discussed in detail in the next section.

8.2 CloudSim Plus Simulation Tool - Brief Overview and Applicability in this Research

CloudSim Plus is an open source toolkit that is developed in Java for simulation of cloud computing infrastructures [56]. As mentioned, it is one of the extended versions of the commonly used open source CloudSim. According to its authors, the modifications made on this version strive to overcome shortcomings of CloudSim. For example, CloudSim Plus has some additional functionalities that provide increased reusability principles [56]. Furthermore, it has an improved class hierarchy that makes it easier to understand. There are also some improvements in the structuring of packages to allow separation of concerns. These changes allow easier reusability, maintainability and extensibility of this toolkit. Based on these, CloudSim Plus allows addition of user defined classes without need to modify core classes. For example, this study adds new class that defines a set of PMs which are initially not available in CloudSim Plus. This is done without modifying core classes that define PMs. In addition, improved class structures also make it easier in this research to modify some of the predefined classes. For example, one of the classes that define datacentre characteristics is modified to include characteristics suitable for this research i.e. the network datacentre is modified to include “power” PMs.

8.2.1 Elements of CloudSim Plus

CloudSim Plus has a number of packages that contain predefined classes used in generation of entities. These entities are independent objects that are generated to create components of the cloud infrastructure [56]. They include, among others, brokers, datacentres, hosts, VMs, VM allocation policies, network devices, and cloudlets (workloads). To briefly describe these entities:

- *Broker:* This is an entity that acts on behalf of cloud users. It helps in submission of user requests which include generation or destruction of VMs.

- *Datacentre:* This is an entity that models the core infrastructure which include both hardware and software components. By default, CloudSim Plus allows generation of only two types of datacentres. These are *power datacentre* and *network datacentre.*

- *Host:* This is an entity that represents PMs in CloudSim Plus datacentres. There are a number of different hosts predefined in CloudSim Plus. They differ based on their resource capacities and datacentre characteristics.

- *VM:* This is an entity that represents virtual instances of operating systems that are submitted to PMs upon user request.
- *Network devices:* These are entities that represent components of network infrastructure. These include different types of switches and network links.
- *VM allocation policies:* These are a set of instructions that represent VM placement algorithm in CloudSim Plus. They provide a mechanism for selection of appropriate PMs for VMs. The default policies of CloudSim Plus are flexible to be replaced with user defined policies.
- *Cloudlets:* These are workloads that are submitted to VMs for execution.

8.2.2 Types of Datacentres in CloudSim Plus

As mentioned, CloudSim Plus allows generation of only two types of datacentres, which are *power datacentre* and *network datacentre.*

Power Datacentre

Power datacentres are used for experiments whereby energy consumption is an expected outcome. These kind of datacentres have no predefined network infrastructure. Instead, they are composed of PMs which are implemented as bins that are ready to accommodate multiple VMs. These PMs implemented in power datacentres are called *power models.* They are explained as the type of PMs that provide energy consumption rates based on hosted workloads [13]. Power models have physical resource configurations such as CPU cores, memory and storage. These physical resource configurations are adjustable to suit requirements of experiments. There are different types of power models available in CloudSim Plus. They differ based on their energy consumption rates per unit workload. Most of these power models simulate actual PMs available in real datacentres.

Power datacentres are generated as either homogeneous or heterogeneous datacentres.

- *Homogeneous Datacentres:* These are the kind of power datacentres that are composed of identical power models.

- *Heterogeneous Datacentres:* These are the kind of power datacentres composed of more than one type of power models.

Network Datacentre

This is a network-aware datacentre used in experiments where network traffic is part of their expected outcome. The components that build up this kind of datacentre are:

1) Network Models: These are PMs that allow connection to network devices. Unlike power models, they have no physical resource configurations. Instead, they consist of *network interfaces* - modules that connect PMs to datacentre network.

2) Network Switches: These are network devices that connect multiple components in the network. In CloudSim Plus, there are different kinds of predefined switches that help in creation of hierarchical networks: core switch, aggregate switch and edge switch.

- *Core Switch* - is a kind of switch that connects entire network to the outside world, or connects different networks together. It is referred to as "level 0" switch in this research.
- *Aggregate Switch* - this is a "level 1" switch that connects core switch to lower levels of the network.
- *Edge Switch* - this is a "level 2" switch that connects directly with PMs. It is intermediate between aggregate switches and PMs.

3) Network Links: These are connection links between network components. They act as passages for transmitting network traffic. They have limited capacities that are adjustable to suit experimental purposes.

The generic structure of network datacentres created in CloudSim Plus is a three layered network. It is composed of all three kinds of switches, connected using network links. The PMs (network models) are connected to edge switches on lowest level of the network (level 2). The network topology for this generic network is shown in Figure 8.1.

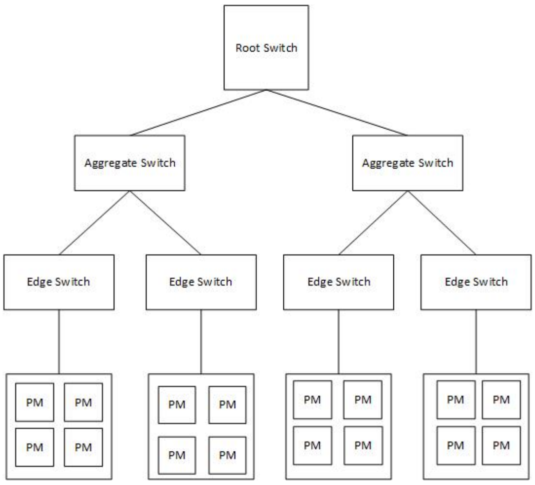


Figure 8.1: Generic Network Topology

CloudSim Plus allows its users to generate different network topologies based on their experiment requirements. For example, a network topology different from generic structure is generated in this research. This topology is generated for a number of datacentres that differ in sizes and configurations. The next section therefore discusses construction and generation of datacentres suitable for evaluation of O-Sec VM placement algorithm.

8.3 Datacentre Architecture - As required for evaluation of O-Sec VM Placement Algorithm

As already discussed, CloudSim Plus supports generation of two types of datacentres, either power datacentres or network datacentres. However, evaluation of O-Sec VM placement algorithm requires characteristics of both datacentres simultaneously. This is because O-Sec VM placement algorithms observe energy consumption minimization and network traffic minimization as part of its objectives. To successfully evaluate this algorithm, this study modifies network datacentre entity predefined in CloudSim Plus to include power datacentre characteristics. This is achieved by introducing a new set of PMs which consist of both network and power model characteristics. Therefore, the existing power models are modified to create new set of PMs with additional feature of network interface. This feature allows compatibility with network topology generated. These new set of PMs replace network models that are initially used in network datacentres.

Also, evaluation of O-Sec VM placement algorithm requires a heterogeneous datacentre. This is to demonstrate PM selection based on their energy consumption rates. In order to generate a heterogeneous datacentre, this study uses two types of power models. These power models differ in energy consumption rates and resource capacities. As mentioned, they are modified to include network model characteristics, while retaining power model characteristics. The used power models are:

- HP ProLiant ML110 G4 (Intel Xeon 3040, 2 Cores x 1840 MHz, 4 GB), referred to as the high capacity PM.

- HP ProLiant ML110 G5 (Intel Xeon 3075, 2 Cores x 2660 MHz, 4 GB), referred to as the low capacity PM.

The network topology used in datacentres generated for evaluations in this study is a three layered network topology. In this network topology, each edge switch has network links that connect to every aggregate switch in the network. This is to allow equal distances between PMs that are connected to different edge switches. In addition, each edge switch connects a combination of both high capacity and low capacity PMs. The topology of this network infrastructure is illustrated using Figure 8.2. Different colors of PMs shown in this figure demonstrate heterogeneity of the datacentre.

To further demonstrate this heterogeneity, Table 8.1 shows energy consumption of PMs connected to the same edge switch. In this table, each of the PMs is allocated a single VM. The table shows that energy consumption of PM0, PM2 and PM4 are lower. These PMs with lower energy consumption are high capacity PMs.

Table 8.1: PM Energy Consumption

Host Name	PM0	PM1	PM2	PM3	PM4	PM5
Energy consumption (W/sec)	57.72	61.12	57.72	61.12	57.72	61.12

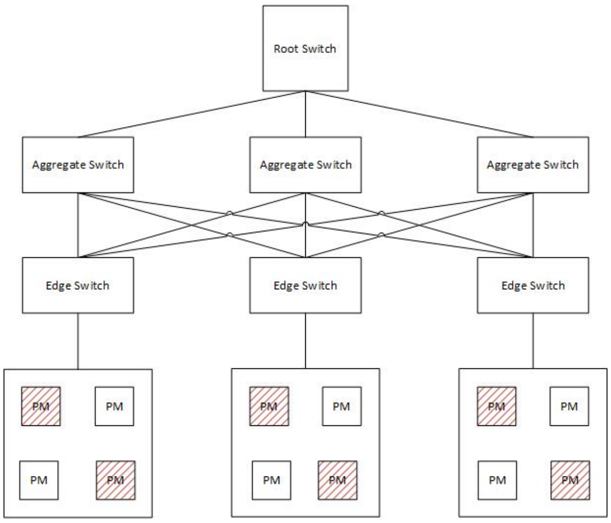


Figure 8.2: Network Topology

8.3.1 Datacentre Configurations

There are three different datacentres created for experiments conducted in this study. The difference between these datacentres is their sizes, which is number of PMs per datacentre. This is to identify the impact brought by different sizes of datacentre on evaluated VM placement algorithms. Configurations for the three datacentres are as follows:

Datacentre 1 Configurations

1. Total Number of PMs = 150
2. Heterogeneous datacentre: PM types = 2
   - HP ProLiant ML110 G4
   - HP ProLiant ML110 G5
3. Network
   - 1 Root switch
   - 3 Aggregate switches (each connects 5 edge switches)
   - 15 Edge switches (each connects 10 PMs)

4. Critical PMs = 3 critical PMs per edge switch

Datacentre 2 Configurations

1. Total Number of PMs = 1000
2. Heterogeneous datacentre: PM types = 2
   - HP ProLiant ML110 G4
   - HP ProLiant ML110 G5
3. Network
   - 1 Root switch
   - 5 Aggregate switches (each connects 20 edge switches)
   - 100 Edge switches (each connects 10 PMs)
4. Critical PMs = 3 critical PMs per edge switch

Datacentre 3 Configurations

1. Total Number of PMs = 4000
2. Heterogeneous datacentre: PM types = 2
   - HP ProLiant ML110 G4
   - HP ProLiant ML110 G5
3. Network
   - 1 Root switch
   - 10 Aggregate switches (each connects 20 edge switches)
   - 200 Edge switches (each connects 20 PMs)
4. Critical PMs = 6 critical PMs per edge switch

8.4 Conclusion

This study uses a simulated cloud environment for evaluations of O-Sec VM placement algorithm. This is because, simulations are less costly, time efficient and allow scalability. The simulation environment of choice in this study is CloudSim Plus. This is because its classes are decoupled in nature, making it easier to modify, extend and maintain. Compared to other simulation environments, the researcher found it easier to use CloudSim Plus for generating environment suitable for evaluations conducted in this study. Having generated a suitable simulation environment, the next step is the evaluation of O-Sec VM placement algorithm. This evaluation process is demonstrated in the next chapter.

Chapter 9

Evaluation

The previous chapter discusses cloud computing simulation environment suitable for evaluations conducted in this study. It further discusses datacentre characteristics and configurations to be used. Having discussed these, the chapter in hand demonstrates evaluation processes of O-Sec VM placement algorithm. It uses discussed cloud computing simulation environment together with datacentre configurations to evaluate this algorithm. The obtained results of O-Sec VM placement algorithm are bench-marked against other related VM placement algorithms, which are worst fit placement algorithm and TPVMP. These VM placement algorithms used in the evaluations of O-Sec VM placement algorithm are discussed in the next section. Other sections demonstrate experiments and analysis made in these evaluations.

9.1 VM Placement Algorithms

As mentioned, evaluations conducted in this chapter include use of worst-fit placement algorithm and TPVMP. In addition, there is also a component of O-Sec VM placement algorithm that is implemented and evaluated which includes only one of the selected security features. This component is named O-Sec 1 VM placement algorithm. Detailed explanations of these algorithms, together with their importance in these evaluations are discussed next.

9.1.1 Worst fit placement algorithm

The worst fit placement algorithm is used as reference to benchmark optimization objectives of O-Sec VM placement algorithms. It is considered as the worst case scenario in terms of VM placement with regard to two optimization objectives observed in O-Sec VM placement algorithm. Therefore, to claim that VM placement algorithm is optimized, the expectation is to obtain traffic costs and energy consumptions that are lower than those of worst fit placement algorithm. To explain this algorithm, it allocates VMs to PMs with most available resources. This means, worst fit placement algorithm considers placement of VMs into idle PMs first. Aggregation of VMs is therefore only possible if all available PMs have at least one hosted VM. This results in higher machine count that causes high energy consumption rates. Worst fit placement algorithm does not consider placement of communicating VMs. Communicating VMs can therefore be placed apart from each other and increase traffic costs. This is further influenced by heterogeneity of the datacentre. It allows placement into PMs with most available resources that are in different locations of the datacentre network. The researcher in this study uses pseudo-code to interpret the implementation of worst fit placement algorithm. This pseudo-code is shown as Algorithm 4.

Algorithm 4 Worst fit algorithm

```
Input:
VM request; Available PMs;
Output:
Allocated PMs;

Description:
Order Available PMs based on available resource capacity;
Allocate VM request in the first Available PM;
IF no Available PMs;
    Reject VM request;
return Allocated PM;
```

9.1.2 TPVMP

This is a VM placement algorithm selected to further be augmented towards the notion of O-Sec VM placement algorithm. It is previously explained as the algorithm that takes into consideration both energy consumption and traffic cost minimization as its objectives. It reduces energy consumption rates by consolidating VMs into high capacity PMs. Also, it reduces traffic costs by placing communicating VMs into PMs that are in close proximity to each other.

9.1.3 O-Sec 1 VM placement algorithm

This algorithm places VMs into available PMs using TPVMP objectives, with an additional security feature identified for O-Sec VM placement algorithm. The security feature of choice in this case is *isolation of critical VMs from non-critical VMs.* This minimizes chances of undesired co-locations that might compromise security of critical VMs. The isolation is achieved by placing critical VMs into PMs dedicated to host them, referred to as critical PMs. Non-critical VMs are also assigned to non-critical PMs. To satisfy TPVMP objectives, the algorithm consolidates VMs into high capacity PMs as much as possible. In addition, it places non-critical VMs into PMs that are in close proximity to their communicating critical VM counter partners. Migrations in this algorithm take into consideration the distance constraint between communicating VMs. They also ensure that critical and non-critical VMs are migrated to appropriate PMs.

9.1.4 O-Sec VM placement algorithm

This is the *final O-Sec VM placement algorithms* that is implemented in this study. It adds more security to the considered optimization and security objectives in O-Sec 1 VM placement algorithm. The additional security considered in this algorithm is the isolation of VMs belonging to adversary users.

Having discussed VM placement algorithms to be used, the next step is experimentation to evaluate and validate O-Sec VM placement algorithm. The next section is the evaluation of the algorithm.

9.2 Evaluation and Validation

There are a number of points that are considered in evaluation of O-Sec VM placement algorithm.

1. VMs used in these experiments are components of the three-tier application. Each VM is a component which directly communicates with two other counter partners (VMs). For every three-tier application, one VM is identified as a critical VM.

2. The traffic cost calculated in these experiments is only internal traffic between communicating VMs. No external traffic cost is calculated.

3. The traffic cost for communicating VMs that are placed within the same PM is zero (0).

4. The traffic cost for communicating VMs that are placed in different PMs which are connected to the same edge switch is one (1).

5. The traffic cost for communicating VMs that are placed in different PMs which are not connected to the same edge switch is two (2).

9.2.1 Experiment 1 (Worst fit placement algorithm and TPVMP)

The aim of this first set of experiments is to demonstrate optimization objectives reflected in TPVMP. This is done by comparing its energy consumption and traffic cost rates with those of worst fit placement algorithm. These are achieved by observing the following in these experiments:

- Energy consumption and traffic costs of VM placement by worst fit placement algorithm.

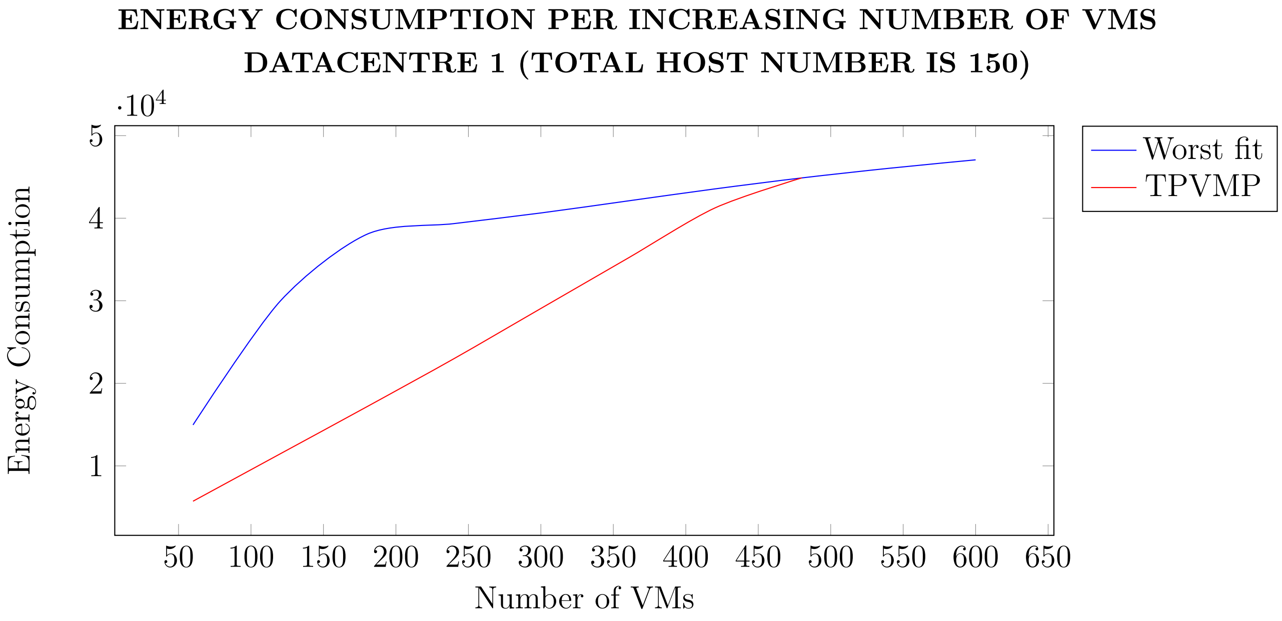
- Comparison between results obtained from placement of VM using worst fit placement algorithm and TPVMP. The expectation is to find lower energy consumptions and traffic costs produced by TPVMP as compared to worst fit placement algorithm, which is *"un-optimized"*.

- Impact made by datacentre sizes on observed optimization objectives of TPVMP.

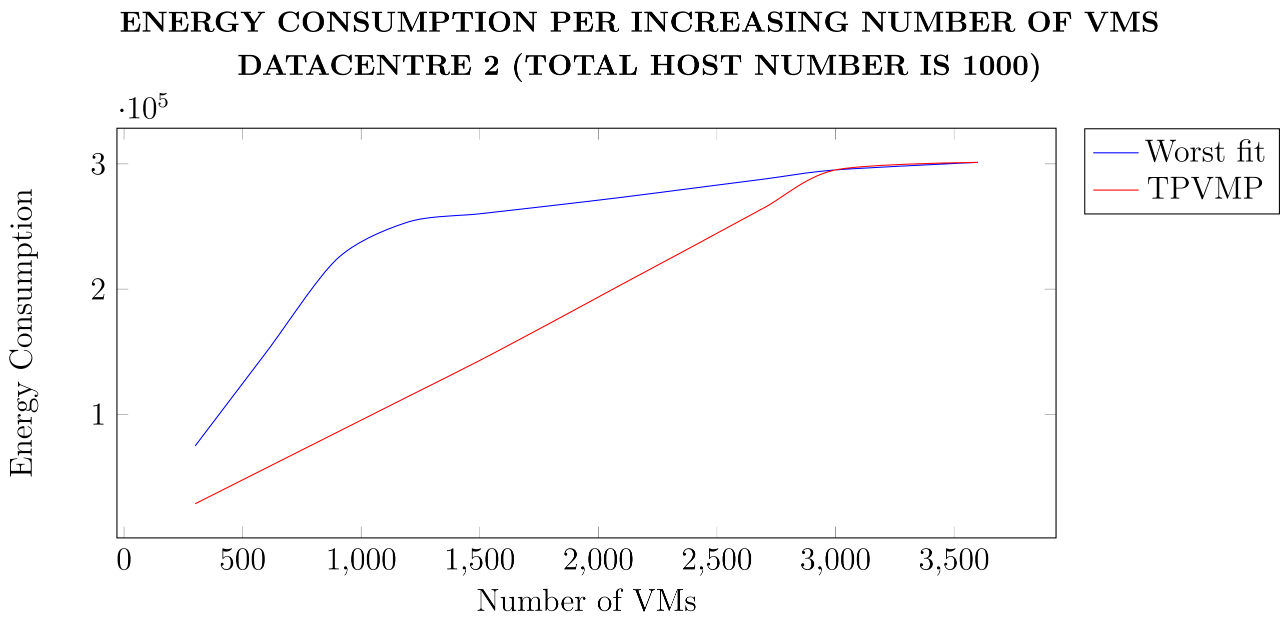
Note: Number of PMs selected to host critical VMs in these datacentre configurations is **3 out of 10** for every edge switch. Each of these identified PMs are high capacity PMs.

Energy Consumptions per increasing number of VMs

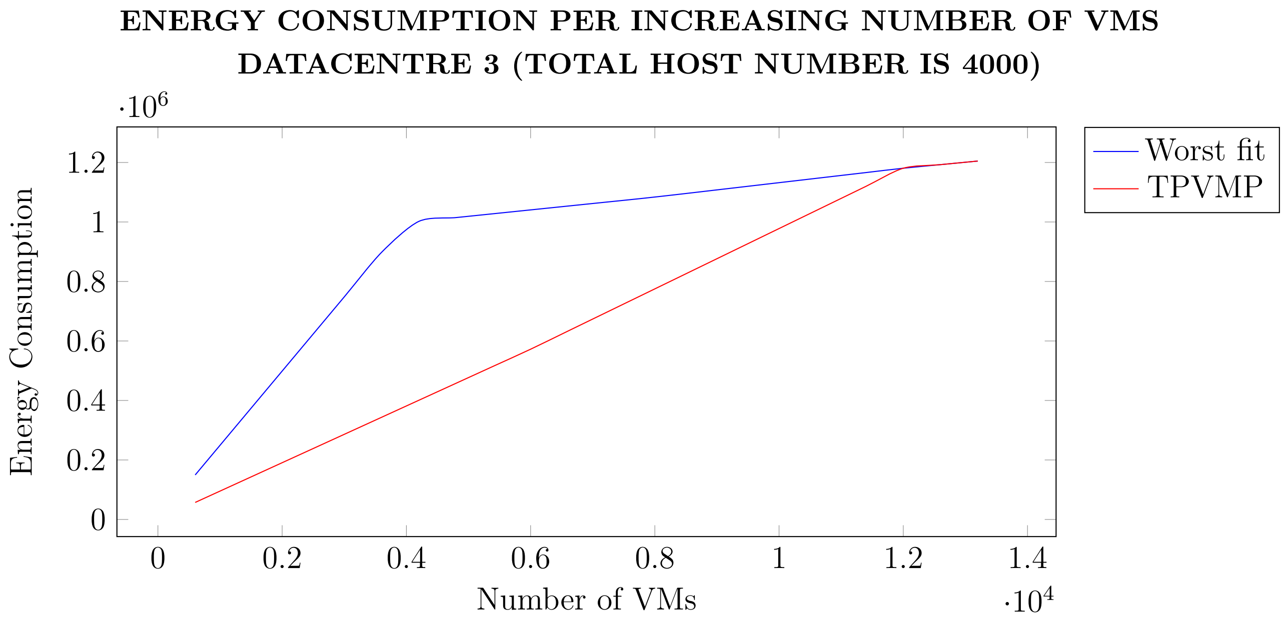
This experiments use three different datacentre configurations to illustrate the results obtained. Graph 9.1, Graph 9.2 and Graph 9.3 show energy consumptions with increasing number of VMs for Datacentre 1, Datacentre 2 and Datacentre 3 respectively.



Graph 9.1: Datacentre 1 - Energy Consumption Rates



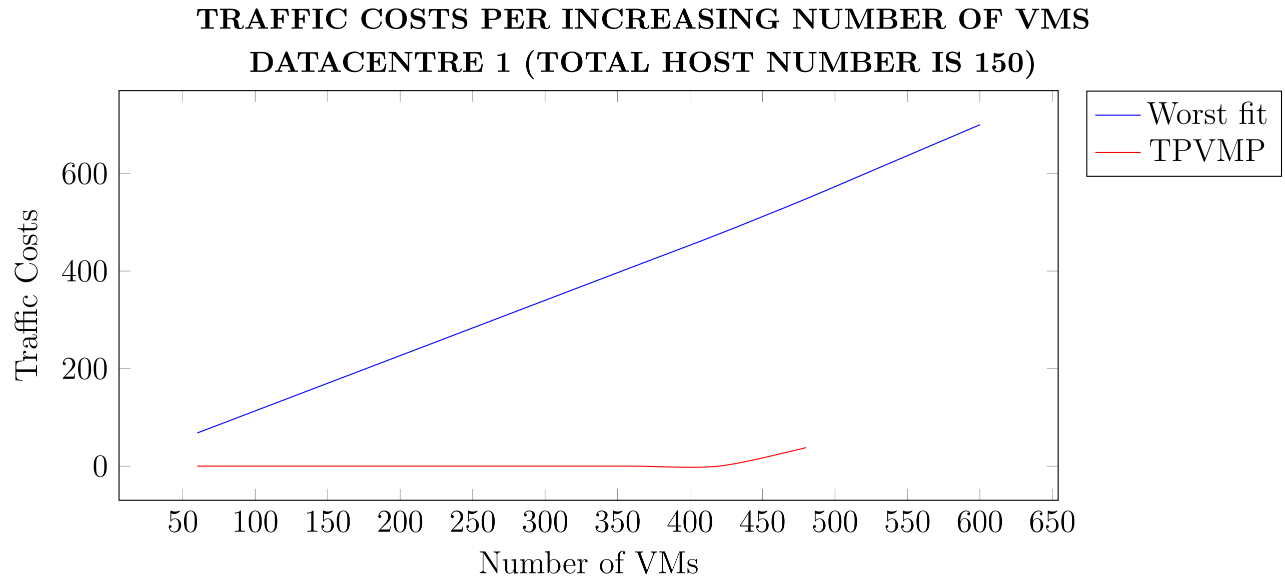
Graph 9.2: Datacentre 2 - Energy Consumption Rates



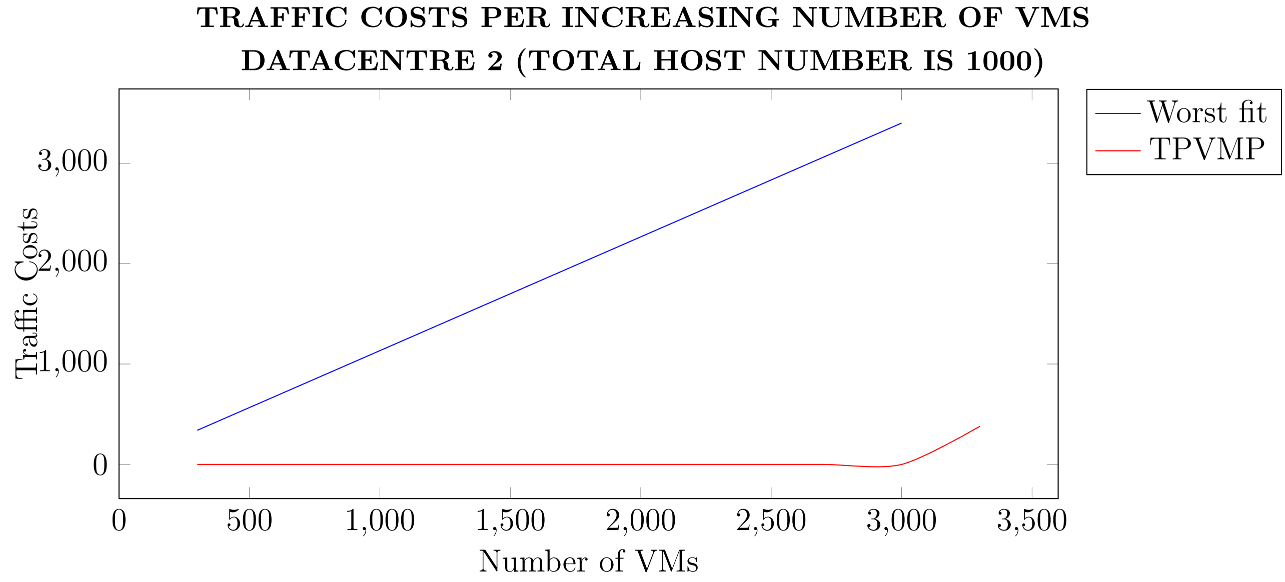
Graph 9.3: Datacentre 3 - Energy Consumption Rates

Traffic costs per increasing number of VMs

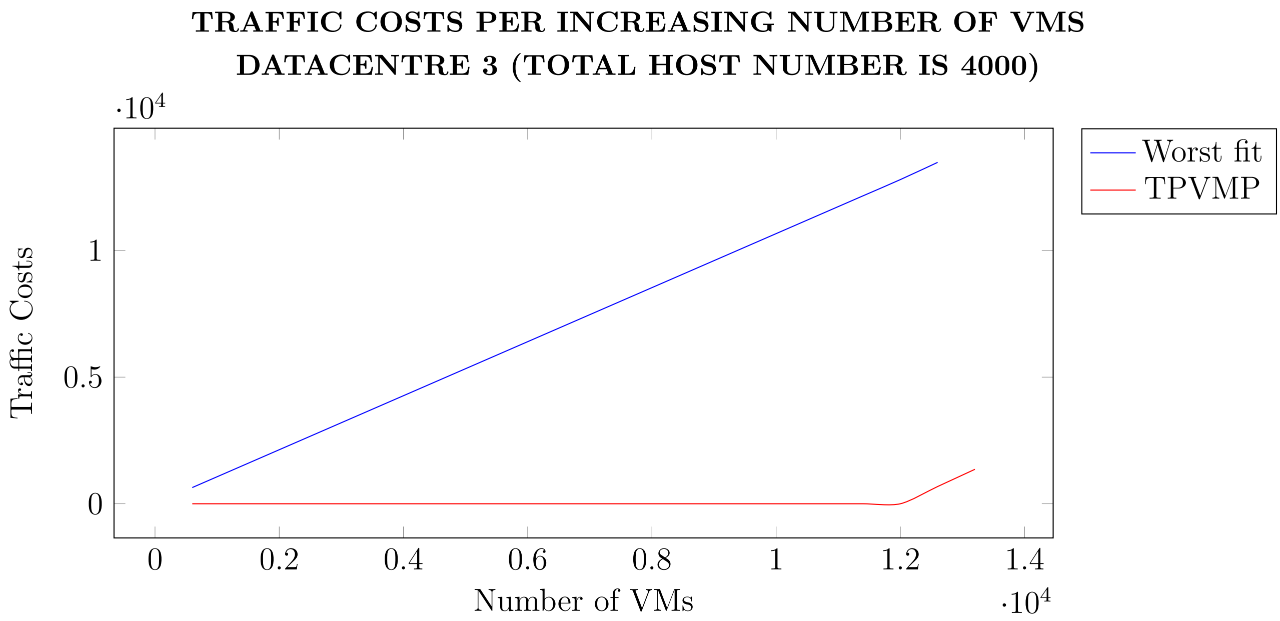
Graph 9.4, Graph 9.5 and Graph 9.6 show traffic costs with increasing number of VMs for Datacentre 1, Datacentre 2 and Datacentre 3 respectively.



Graph 9.4: Datacentre 1 - Traffic costs



Graph 9.5: Datacentre 2 - Traffic costs



Graph 9.6: Datacentre 3 - Traffic costs

Observations

Based on results shown in the displayed graphs (Graph 9.1 - 9.6), the following analysis are made:

Worst fit placement algorithm

Energy Consumption: The increasing number of VMs placed using worst fit placement algorithm results in increasing energy consumption. These increasing rates are higher in the beginning of VM placement, but subside as the number of VMs increase. This is influenced by heterogeneity of generated datacentres. The low capacity PMs in these datacentres have more physical resource capacities as compared to high capacity PMs. This therefore results in placement into low capacity PMs first, hence the rapid increase in energy consumptions.

Traffic cost: Traffic costs produced by placement of VMs using worst fit placement algorithm increase in proportion to the increasing number of VMs.

Worst fit placement algorithm vs. TPVMP

Energy Consumption: As it is expected, energy consumptions produced by TPVMP are lower compared to those of worst fit placement algorithm. However, this is true only when datacentre is not fully utilized. For a number of VMs approaching datacentre maximum capacity, energy consumption produced by TPVMP approaches that produced by worst fit placement algorithm.

Traffic cost: TPVMP traffic costs remain zero at lower number of VMs placed. However, it rises from zero and grows proportionally with increasing number of VMs at a higher number of VMs (approaching datacentre capacity). This is because when there are enough resource capacities, communicating VMs are placed in same PMs. With limited resource capacities left in all PMs, some of these communicating VMs are placed in different PMs where they fit.

Datacentre size impact

There are insignificant variations in energy consumption and traffic cost patterns of three datacentres. This means that the size of datacentres has little impact, if any, to the results obtained in this study. It is therefore up to the researcher to decide on which datacentre to use in order to evaluate and validate O-Sec VM placement algorithm.

General Comments

Results show that TPVMP is completely optimized at lower number of VMs. That is, energy consumption and traffic costs are on their lowest at fewer number of VMs placed in the datacentre. Other observations show that any of the three datacentres can be used for experiments. This is because, sizes of datacentres bring insignificant variations to the results obtained from these experiments. Based on this, the researcher in this study decides to use Datacentre 2 (Total number of hosts is 1000) for the rest of these experiments. This datacentre is chosen because it takes reasonable time to run experiments with higher number of VMs as compared to Datacentre 3.

9.2.2 Experiment 2 (O-Sec 1 VM placement algorithm)

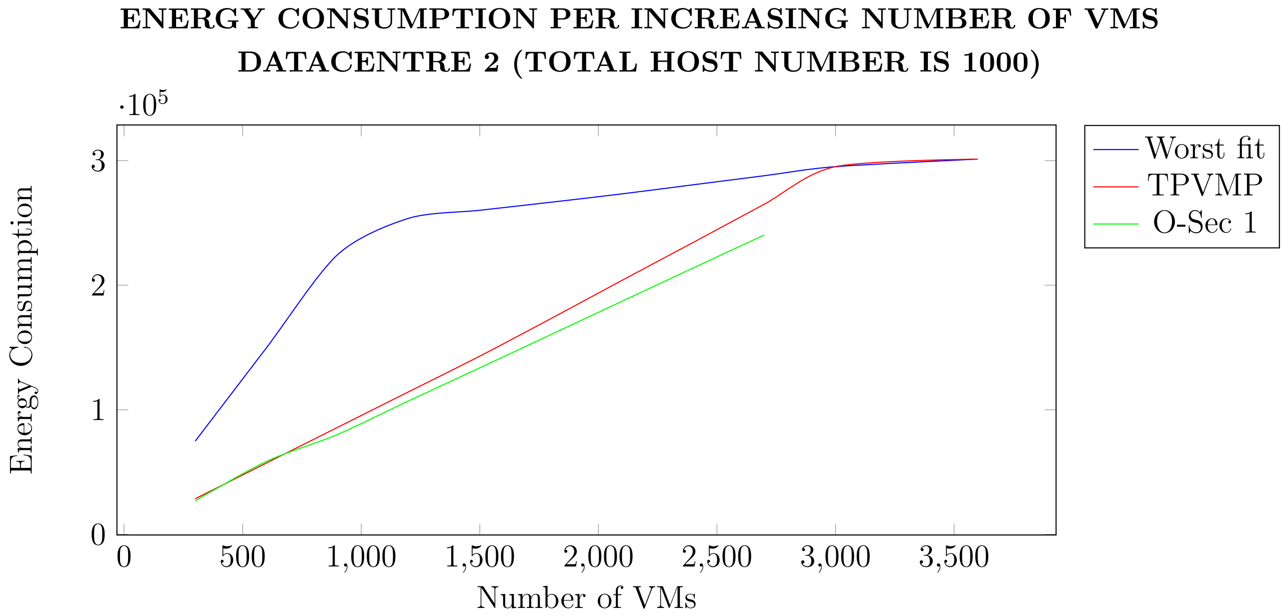
The aim of this set of experiments is to observe whether optimization is maintained in implementation of O-Sec 1 VM placement algorithm. Furthermore, it observes the impact of enforced security features on the already existing optimization objectives.

These are achieved through the following:

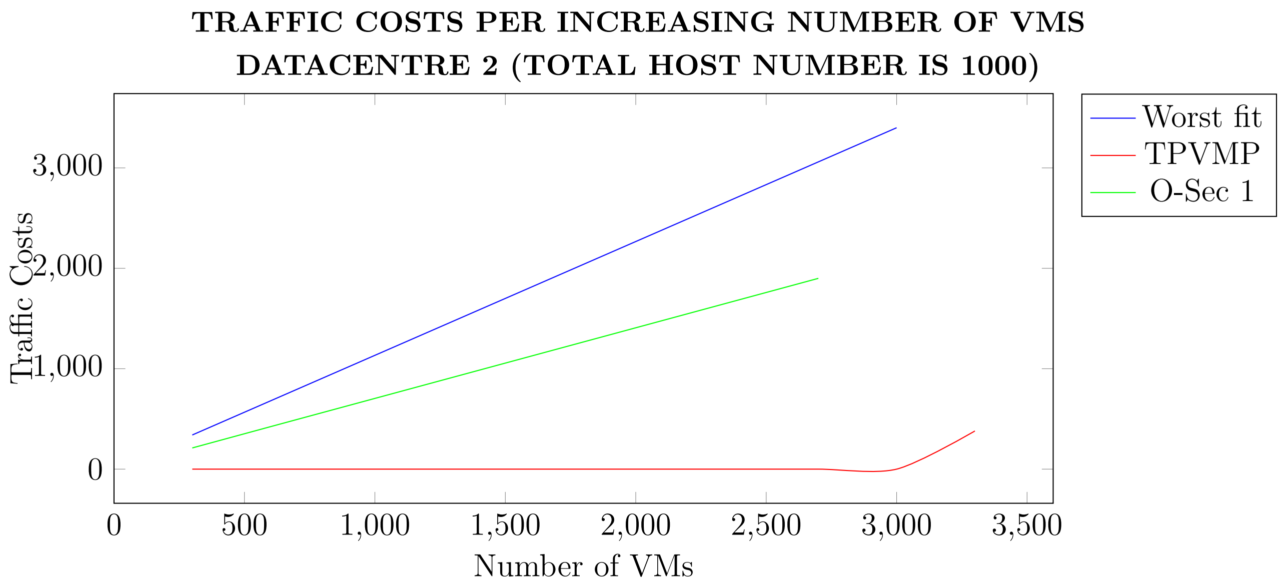
- Comparison of optimization objectives between results obtained from placement of VMs using O-Sec 1 VM placement algorithm and worst fit placement algorithm.
- Comparison of optimization objectives between results obtained from placement of VMs using O-Sec 1 VM placement algorithm and TPVMP.

The following graphs (Graphs 9.7 - 9.8) show results obtained from this set of experiments.

Note: Number of PMs selected to host critical VMs in these datacentre configurations is **3 out of 10** for every edge switch. Each of these identified PMs are high capacity PMs.



Graph 9.7: O-Sec 1 VM placement algorithm - Energy Consumption Rates



Graph 9.8: O-Sec 1 VM placement algorithm - Traffic costs

Observations

Based on results shown in Graph 9.7 and Graph 9.8, the following analysis are made:

O-Sec 1 VM placement algorithm vs. Worst fit placement algorithm
Energy Consumption: According to Graph 9.7, there are significant differences in the amounts of energy consumptions produced by placement of VMs using the two algorithms. Energy consumptions for O-Sec 1 VM placement algorithm are significantly lower compared to those of Worst fit placement algorithm.

Traffic cost: According to Graph 9.8 the traffic costs produced by VMs placed using O-Sec 1 VM placement algorithm are lower compared to those of worst fit placement algorithm.

In general, both these observations show that optimization is retained in the implementation of O-Sec 1 VM placement algorithm.

O-Sec 1 VM placement algorithm vs. TPVMP
Energy Consumption: According to Graph 9.7, the energy consumption increase rates

are almost the same at lower number of VMs (below 700 VMs placed). However, there is an unexpected observation made for number of VMs above 900. This is, energy consumption rates produced by O-Sec 1 VM placement algorithm are lower than those produced by TPVMP. From researcher's analysis, these results are influenced by selection of critical PMs as high capacity PMs. To explain this, TPVMP on one hand strives to consolidate VMs into PMs that are in close proximity to each other, starting with high capacity PMs. However, if there are limited resources in high capacity PMs, then the algorithm places VMs into low capacity PMs that are in close proximity to those hosting their communicating partners. This results in utilization of one leave of the network at a time where both high capacity and low capacity PMs are used. O-Sec 1 VM placement algorithm on the other hand, places critical VMs into PMs selected to host them. These PMs, which are limited in every edge switch, are high capacity PMs. The algorithm further places the communicating counter partners of the critical VMs into PMs that are in close proximity. Due to limited number of PMs selected to host critical VMs in every edge switch, critical VMs are spread across different leaves of the datacentre network. Therefore, communicating counter partners exhaust high capacity PMs available in different leaves of the network before occupying low capacity PMs. In general, O-Sec 1 VM placement algorithm strives to exhaust high capacity PMs across the entire network before selecting low capacity PMs. But, TPVMP uses PMs connected to the same edge switch, and only moves to the next if resources are exhausted. This means TPVMP uses more of the low capacity PMs than O-Sec 1 VM placement algorithm, hence the higher energy consumption.

Traffic cost: According to Graph 9.8, traffic costs produced by O-Sec 1 VM placement algorithm are higher compared to those produced by TPVMP. They increase with increasing number of VMs placed. This is because for every placement, communicating critical VMs are separated from their non-critical counter partners.

Datacentre Capacity

O-Sec 1 VM placement algorithm limits capacity of datacentres. This is observed from shorter graphs produced by placements using this algorithm. The limitation is influenced by fewer number of PMs identified to host critical VMs. If these PMs are exhausted, new VM requests for placement of critical VMs are rejected.

Comments

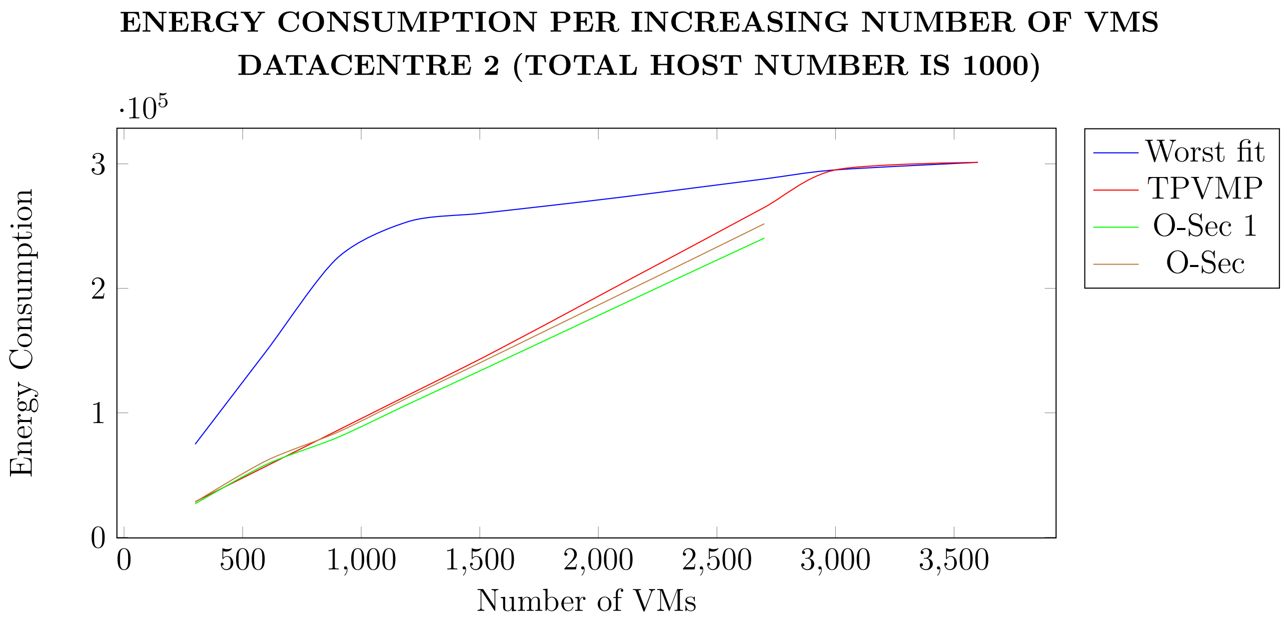
According to the observations, it is correct to claim that O-Sec 1 VM placement algorithm is completely optimized. This is true because both its energy consumptions and traffic costs are relatively low compared to those of worst fit placement algorithm. However, added security feature (isolation of critical from non-critical VMs) makes an impact to the optimization objectives observed in O-Sec 1 VM placement algorithm. That is, energy consumption rates decrease slightly and traffic costs increase significantly as compared to those of TPVMP. Lastly, the limited number of PMs selected to host critical VMs limits capacity of datacentres.

9.2.3 Experiment 3 (O-Sec VM placement algorithm)

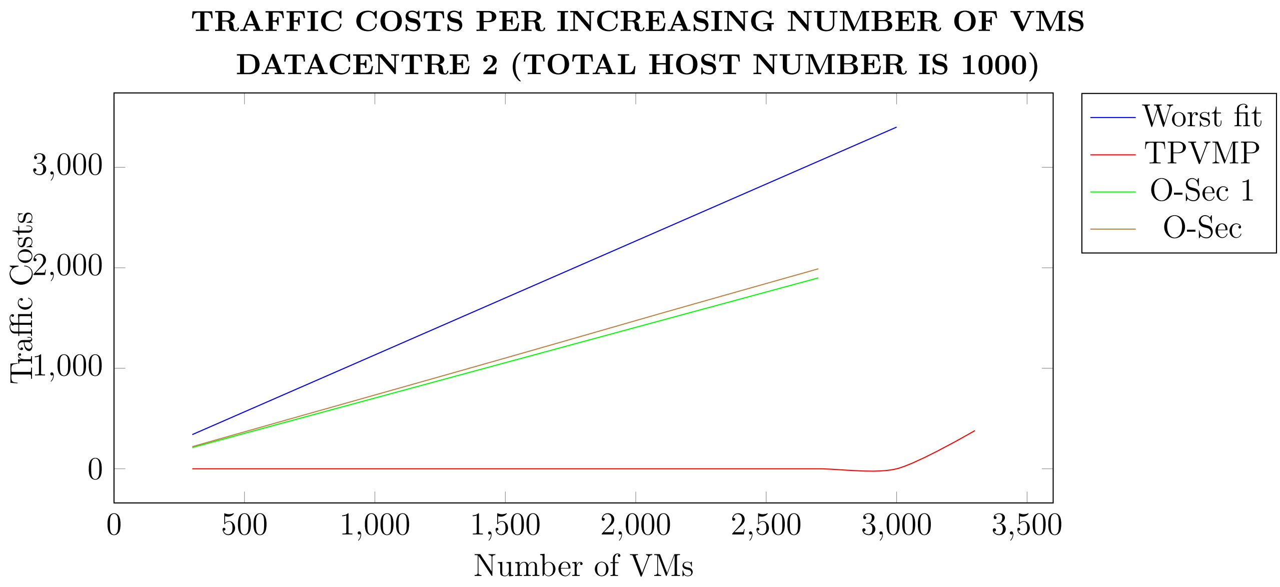
This set of experiments focus on final O-Sec VM placement algorithm implemented in this study. The aim is to observe if optimization objectives of TPVMP are maintained after enforcing security features. In addition, these experiments also observe the impact of security features to the overall behaviour of optimization objectives. The observations are achieved through the following:

- Comparison of optimization objectives between results obtained from placement of VMs using O-Sec VM placement algorithm and worst fit placement algorithm.
- Comparison of optimization objectives between results obtained from placement of VMs using O-Sec VM placement algorithm and TPVMP. This is to observe the impact of the implemented security features on these optimization objectives.

The following graphs (Graphs 9.9 - 9.10) show results obtained from this set of experiments.
Note: Number of PMs selected to host critical VMs in these datacentre configurations is **3 out of 10** for every edge switch. Each of these identified PMs are high capacity PMs.



Graph 9.9: O-Sec VM placement algorithm - Energy Consumption Rates



Graph 9.10: O-Sec VM placement algorithm - Traffic costs

Observations

Based on the results shown in Graph 9.9 and Graph 9.10, the following observations are made:

O-Sec VM placement algorithm vs. Worst fit placement algorithm

Energy Consumption: According to Graph 9.9, there are significant differences in the amounts of energy consumptions produced by placement of VMs using two algorithms. Energy consumptions of O-Sec VM placement algorithm are significantly lower compared to those of worst fit placement algorithm.

Traffic cost: According to Graph 9.10, traffic costs produced by O-Sec VM placement algorithm are lower compared to those produced by worst fit placement algorithm.

In general, both these observations show that optimization is retained in implementation of O-Sec VM placement algorithm.

O-Sec VM placement algorithm vs. TPVMP

Energy Consumption: As it is the case with O-Sec 1 VM placement algorithm, Graph 9.9 shows lower energy consumption rates for O-Sec VM placement algorithm relative to those of TPVMP.

Traffic cost: According to Graph 9.10 traffic costs produced by O-Sec VM placement algorithm are higher compared to those produced by TPVMP. Like those of O-Sec 1 VM placement algorithm, they increase with increasing number of VMs placed.

O-Sec VM placement algorithm vs. O-Sec 1 VM placement algorithm

Energy Consumption: According to Graph 9.9, increase rates of energy consumptions produced by O-Sec VM placement algorithm are slightly higher compared to those produced by O-Sec 1 VM placement algorithm.

Traffic cost: According to Graph 9.10 traffic costs produced by O-Sec VM placement algorithm are slightly higher compared to those produced by O-Sec 1 VM placement algorithm.

Datacentre Capacity

As it is the case with O-Sec 1 VM placement algorithm, capacity of datacentre is limited by fewer number of PMs identified to host critical PMs.

Comments

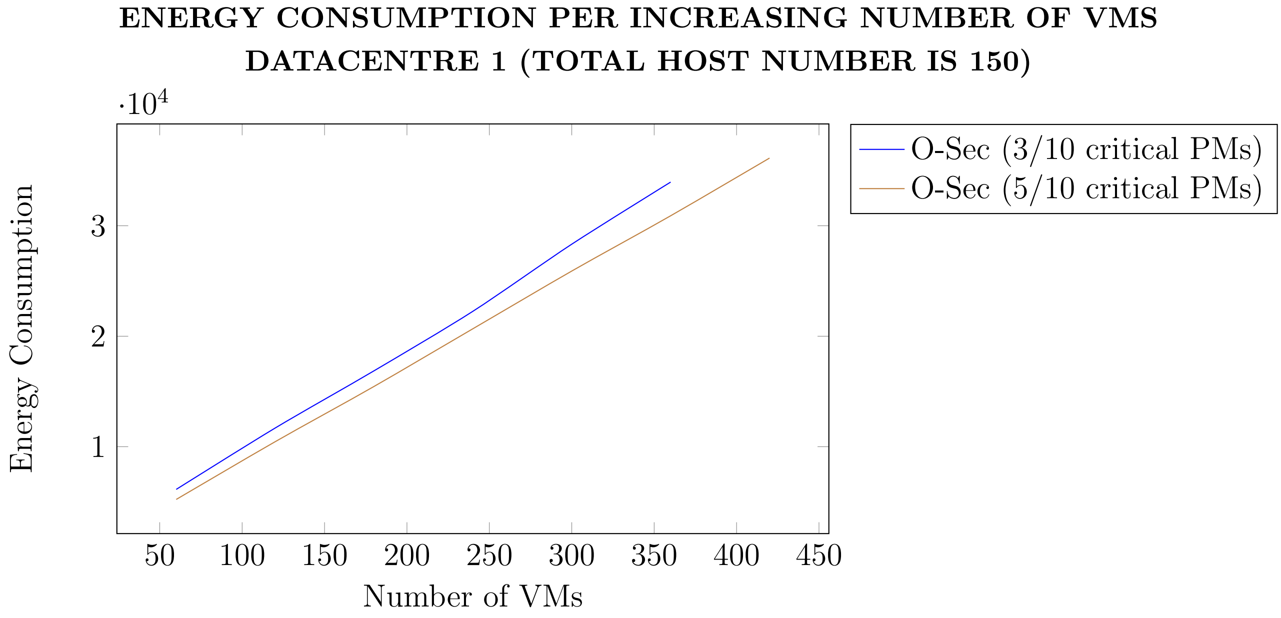
O-Sec VM Placement algorithm is completely optimized because both its energy consumptions and traffic costs are relatively low compared to worst fit placement algorithm. However, added security features make an impact to optimization objectives observed. Energy consumption rates decrease slightly as compared to those of TPVMP, while traffic costs increase significantly. In addition, the limited number of PMs selected to host critical VMs limits the capacity of datacentre.
Compared to O-Sec 1 VM placement algorithm, energy consumptions and traffic costs produced by O-Sec VM placement algorithm increase slightly with increasing number of VMs. This means, an additional security feature has an impact on placement of VMs into the cloud infrastructure. However, these impact does not jeopardize optimization objectives.

9.2.4 Experiment 4 (Varying number of critical VM hosts)

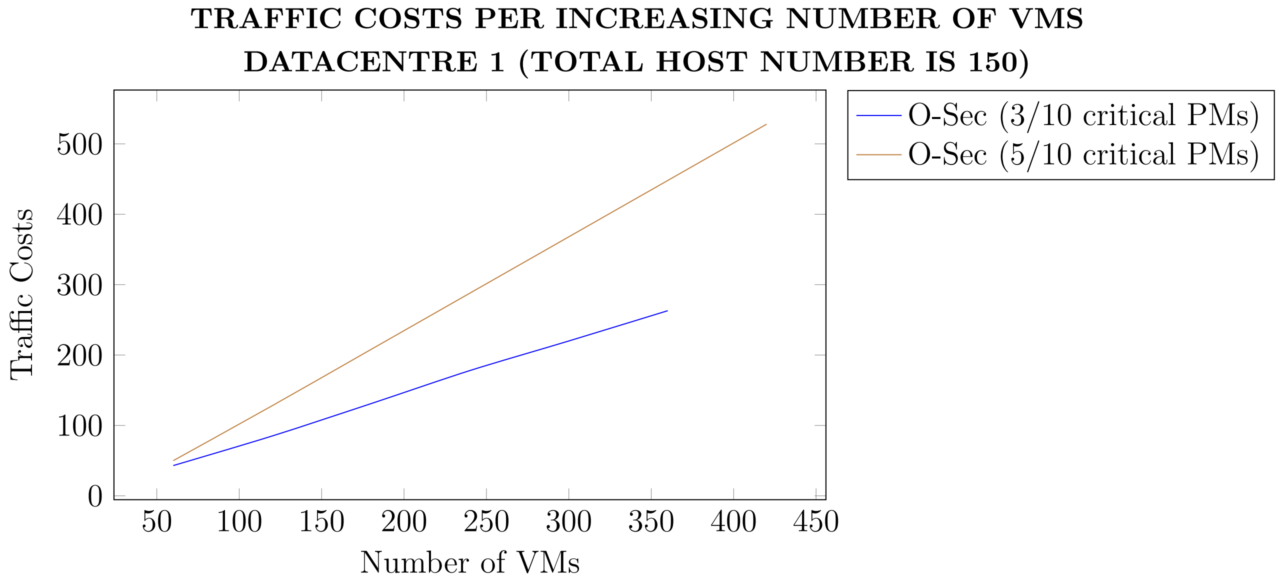
This set of experiments aim to observe an impact made on O-Sec VM placement algorithm results when varying number of PMs to host critical VMs. As seen in previous experiments, limited number of these PMs limits capacity of datacentres. For these experiments, the researcher aims to observe the impact brought by varying number of PMs to host critical VMs. This is achieved through the following:

- Comparison of optimization objective results obtained when number of PMs to host critical VMs is varied.
- Comparison of datacentre capacities when number of PMs to host critical VMs is varied.

Configurations of the three datacentres previously created comprises of 3 out of 10 PMs identified to host critical VMs. This number is varied in this set of experiments in order to observe its impact on O-Sec VM placement algorithm objectives. The number of PMs identified to host Critical VMs in this case are increased to 5 out of 10 for every edge switch. Datacentre 1 configurations are used in this experiments because its smaller size makes it easier to reconfigure and obtain results. The following graphs (Graphs 9.11 - 9.12) display results obtained after increasing number of PMs to host critical VMs.



Graph 9.11: Varying Number of Critical PMs - Energy Consumption Rates



Graph 9.12: Varying Number of Critical PMs - Traffic costs

Observations

Energy Consumption: Graph 9.11 shows that energy consumption rates of O-Sec VM placement algorithm are lower for higher number of PMs selected to host critical VMs.

Traffic Cost: Graph 9.12 shows that traffic costs increase significantly with increasing number of PMs to host critical VMs.

Datacentre Capacity: Lastly, increasing number of critical PMs results in increase to capacity of datacentres. Graph 9.11 and Graph 9.12 show longer graphs for increased number of PMs. For a 3/10 number of critical PMs, capacity shown for O-Sec VM placement algorithm is about 360 VMs. But, for 5/10 number of critical PMs the capacity shown is about 420 VMs.

Comments

Generally, increasing number of critical PMs within datacentre results in rapid increase to traffic costs. In addition, capacity of the datacentre increases with increasing number of critical PMs. This means, there is a trade-off between capacity of datacentres and traffic costs when identifying number of critical PMs in the datacentre.

9.2.5 Analysis

This sub-section aims to provide researcher's perspective on results obtained from the conducted experiments. From the observations made in each of these experiments, the researcher made note of the following important points:

- *Enforced security in O-Sec VM placement algorithm affects optimization objectives inherited from TPVMP*

This conclusion is made from the following observations:
1) Energy consumptions produced by O-Sec VM placement algorithm are lower than those produced by TPVMP. As mentioned, this unexpected behaviour is influenced by limited number of critical PMs in every leave of the network, which allow critical VMs to be placed across the entire network. Based on distance constraint observed in O-Sec

VM placement algorithm and TPVMP, communicating counter partners of these critical VMs are placed in PMs that are in close proximity. This results in all VMs being spread across the entire network, and placed in high capacity PMs first. It therefore results in lower energy consumptions as compared to TPVMP.

2) Traffic costs produced by O-Sec VM placement algorithm are significantly higher than those produced by TPVMP. This is because TPVMP places communicating VMs in close proximity to each other. It firstly consolidates communicating VMs into same PMs. However, if there are limited resources, the algorithm places communicating VMs into PMs that are in close proximity. This results in very low traffic costs, which remain zero at lower number of VMs placed. The results differ for O-Sec VM placement algorithm because the algorithm separates communicating VMs. The separation mainly results from isolation of critical from non-critical PMs. So, for every set of communicating VMs, critical VM is separated from its counter partners. It results in increased traffic cost per placement of communicating VMs. Furthermore, it is influenced by isolation of VMs belonging to adversary users and the limited number of PMs to host critical VMs in every network leave.

- *Varying number of PMs that are assigned to host critical VMs affect optimization objectives and datacentre capacities*

1) Energy consumption of O-Sec VM placement algorithm decreases with increasing number of PMs to host critical VMs. However, this increase is not significant, more especially at lower number of VMs placed. This means that varying the number of PMs to host critical VMs affects the energy consumptions slightly.

2) Traffic cost of O-Sec VM placement algorithm increases rapidly with increasing number of PMs assigned to host critical PMs. This means higher number of PMs assigned to host critical PMs put in jeopardy QoS. Although it limits the capacity of datacentre, it is advisable to keep this number of PMs to host critical VMs lower.

- *O-Sec VM placement algorithm is optimized*

The optimization objectives inherited from TPVMP are retained in the implementation of O-Sec VM placement algorithm. This is true because energy consumptions and traffic costs produced by placement using O-Sec VM placement are significantly lower than those produced by worst fit placement algorithm. Although, the enforced security objectives affect these optimization objectives negatively, the change is acceptable and does not completely jeopardize optimization.

9.3 Conclusion

From analysis made in this chapter, it is safe to conclude that O-Sec VM placement observes both *security* and *optimization* objectives. This is true because, on one hand, security features are enforced in implementation of this algorithm. The researcher ensures that this algorithm isolates critical VMs from non-critical VMs. In addition, he also ensures that VMs belonging to adversary users are isolated from each other. On the other hand, results show that optimization objectives inherited from TPVMP are retained in O-Sec VM placement algorithm. Although security features of O-Sec VM placement algorithm affect these optimization objectives, the impact is only slight. This is true because optimization objectives remain significantly lower than those of worst fit placement algorithm.

However, there is an observation made which affects optimization of O-Sec VM placement algorithm. This is, number of PMs selected to host critical VMs affects optimization objectives of O-Sec VM placement algorithm. It is discovered that higher number of these PMs results in high traffic costs thus affecting QoS expected in placement by O-Sec VM placement algorithm. It is therefore advisable to keep number of PMs to host critical VMs low.

Chapter 10

Conclusions

This study discussed minimization of inter-VM attacks that are caused by undesired co-location of VMs. It proposed a security-aware VM placement algorithm that strives to minimize these undesired co-locations while providing good QoS at reduced operational costs. The aim was to ensure secure placement of VMs that does not jeopardize performance of the cloud infrastructure. The prototype of this proposed VM placement algorithm was implemented and evaluated in previous chapters.

This chapter revisits the problem statement and research question. It further evaluates whether this study achieves the aim of this research. Finally, it stipulates the main contributions of this research and make recommendations for future work.

10.1 Revisiting the problem statement

The main focus of this study was to implement an optimized security aware VM placement algorithm. The aim was to enhance security in placement of VMs while achieving good QoS at reduced costs. The solution therefore ensured security with at least one or more of the objectives classified under optimization. This was achieved by enhancing an existing and optimized VM placement algorithm with security features. Two security features were used to enhance the selected VM placement algorithm, which are: isolation of VMs belonging to adversary users, and isolation of critical VMs from non-critical

VMs. These security features strive to minimize inter-VM attacks which are common in multi-tenant cloud computing environments.

The main claim of this study was formulated as follows:
Both security and optimization objectives can be observed in implementation of VM placement algorithms.

10.1.1 Answering the research Questions

The main goal of this research was to answer the following main question:

How can optimized security-aware VM placement algorithms be designed, developed and implemented without jeopardizing optimization objectives that ensure good quality of services, at reduced costs and without violating SLA?

To successfully answer this main question, there were a number of fundamental questions that were stipulated in the first chapter as research sub-questions. These sub-questions, and how they were addressed in this study, are discussed as follows:

- *How is cost reduction and QoS achieved in the implementation of VM placement algorithms?*

This required thorough literature review of the implemented VM placement algorithms. Furthermore, it required identification and in-depth study of VM placement algorithms that strive towards providing good QoS and/or minimizing operational costs. This included studies on the challenges associated with cost reduction and QoS objectives. To address this sub-question, Chapter 3 focussed on literature of available VM placement algorithms. It further classified these VM placement algorithms into three categories: VM placement algorithms that focus on QoS, VM placement algorithms that focus on cost reduction and VM placement algorithms that focus on security. An in-depth study of each of these categories was made. It included an overview on implementations of different VM placement algorithms classified under each of the mentioned categories. In addition, shortcomings caused by these VM placement algorithms and their possible countermeasures were discussed.

- *How can cost reduction and QoS be integrated into security-aware VM placement algorithms?*

To successfully implement a VM placement algorithm that includes cost reduction, QoS and security, a particular approach was identified. An identified approach was selection of an already existing VM placement algorithm for augmentation towards the notion of O-Sec VM placement algorithm. An evaluation criteria was used in chapter 6 for selection of an appropriate VM placement algorithm. This criteria, which used quantitative method, selected the VM placement algorithm that assumed highest objectives, regardless of their category. The selected VM placement algorithm which scored highest in those evaluations was TPVMP. The researcher in this study decided to use two different security features to augment the selected VM placement algorithm. These are: isolation of VMs belonging to adversary users, and isolation of critical from non-critical VMs.

- *What are security requirements for O-Sec VM placement algorithms?*

This required literature review of most common security risks available in cloud computing. It further required identification of the possible vulnerabilities that cause these risks. The identified vulnerabilities are those brought by cloud computing characteristics discussed in Chapter 2. These characteristics bring, among others, vulnerabilities associated with undesired co-location of VMs. They include co-location of VMs belonging to adversary users and co-location of VMs with different vulnerability statuses. The identified countermeasures for these possible vulnerabilities were discussed in Chapter 7. These countermeasures contributed towards selection of appropriate security features used for implementation of O-Sec VM placement algorithm.

- *What are hypervisors and how are they involved in placement of VMs within the cloud?*

This also required a thorough literature review of hypervisors. It included review on benefits brought by use of hypervisors. Also, how they contribute to the placement of VMs within cloud infrastructure. To achieve this, Chapter 2 discussed hypervisors and their benefits. Those benefits include isolation, resource sharing and VM placement capabilities. Among the discussed hypervisor benefits, VM placement capabilities is unique to this study. It supports cloud computing characteristics associated with VM life cycle. These supported characteristics are: on-demand self-service, utility, self-managing and broad network access.

- *How are hypervisors changed to allow VM placement that includes cost reduction, QoS and security?*

Ideally, this required identification of hypervisor components that contribute towards placement of VMs. It further required replacement of the identified hypervisor components in order to change the default placement strategy. This study used a simulated cloud environment to run experiments. This means, default placement component replaced was that of the cloud simulation tool.

Having answered research sub-questions, the next step is to discuss how the main objective was achieved. This main objective combines answers to the sub-questions in order to build answers to the main question. The next sub-section discusses how the main objective of this study is achieved.

10.1.2 How is the main objective achieved?

The research objective of this study was achieved through completion of a number of phases. Each of these phases was a partial solution to the discussed research sub-questions. To discuss these phases:
Through *literature review*, the study identified possible vulnerabilities that are brought by cloud computing characteristics. It specifically concentrated on vulnerabilities brought by virtualization and/or aggregation of VMs. Countermeasures against risks brought by these kinds of vulnerabilities were discussed. They involved secure placement of VMs using hypervisor component referred to as VM placement algorithm. It is important to note that cloud computing characteristics were revised in this study. The revised list of these characteristics was further used to re-define cloud computing. This was to ensure that the definition of cloud computing suited the purpose of this study.

The gathered literature helped in identification of the most relevant security features required in the *design, development and implementation* of O-Sec VM placement algorithm. These security features were built from discussed countermeasures which involved isolation of VMs to minimize risks brought by undesired co-locations. An approach taken to design, develop and implement O-Sec VM placement algorithm was to transform an existing VM placement algorithm. To achieve this, the study identified a VM placement algorithm that assumed highest optimization and/or security objectives. It then augmented this algorithm with identified security features in an incremental manner. This incremental manner was used to allow observations on the VM placement algorithm behaviour when security features are added.

Some *evaluation criteria* was used to find the most suitable VM placement algorithm to be transformed towards O-Sec VM placement algorithm. This evaluation criteria involved four (4) steps which started with elimination of irrelevant VM placement algorithms. This was done through abstract reading and literature review of VM placement algorithms publications. Next, the criteria identified the most important optimization and security objectives used for the evaluation process. These objectives were rated based on their importance, with rating ranging from 1 (necessary) to 4 (critically important). The third step was scaling of the existence of these identified objectives in each of the selected VM placement algorithms. The scale ranged from 0 (does not exist) to 2 (exists fully). The last step used quantitative method to calculate percentage score of the reflected objectives in each of the selected VM placement algorithms. It combined ratings of objectives together with scaling of their existence and used a formula to calculate percentage scores. The highest scoring VM placement algorithm (TPVMP) was selected as an appropriate VM placement algorithm to further be augmented towards O-Sec VM placement algorithm. TPVMP observes two optimization objectives, which are: traffic cost minimization and energy consumption minimization. Fortunately, these objectives allowes it to be categorized under two-out-of-three VM placement algorithms categories. These are VM placement algorithms that focus on cost reduction and VM placement algorithms that focus on QoS. Therefore, augmenting TPVMP with security features makes it stretch across all VM placement categories discussed in this study. As mentioned, TPVMP was augmented with two security features in order to implement

O-Sec VM placement algorithm.
To successfully *evaluate and validate* O-Sec VM placement algorithm, a cloud simulation tool was used instead of real infrastructure. This was because experiments conducted on real infrastructures are costly, time consuming and limits scalability factor. The cloud simulation tool of choice for this study was CloudSim Plus. It was chosen among other cloud simulation tools because of its unique characteristics. These characteristics include: structured packages that allow separation of concerns, functionalities that provide reusability principles, and improved class hierarchy that makes its code easier to understand. These characteristics together make this tool easier to reuse, maintain and extend as compared to others. In these evaluation and validation processes, two algorithms were used to benchmark results obtained from O-Sec VM placement algorithm. These were worst fit placement algorithm and TPVMP. Worst fit placement algorithm, on one hand, was used as threshold to determine whether an O-Sec VM placement algorithm retained the optimization objectives. TPVMP, on the other hand, was used to observe the impact of added security features to the existing optimization objectives inherited from TPVMP itself. The inherited objectives are energy consumption minimizations and traffic cost minimizations.

10.2 Main Contributions

10.2.1 Re-defining Cloud Computing

This research re-defined cloud computing based on its characteristics. It revisited characteristics that were previously discussed in the existing cloud computing publications. Furthermore, it changed and/or modified some of the characteristics in order to suit its objective. There were newly defined characteristics that are unique to this study. However, some of these characteristics exist in other publications, but were renamed to broadly cover cloud computing characteristics that were overlooked in previous studies. The newly defined cloud computing characteristics are:

- **Utility**: It shows that cloud computing services are provisioned based on user demands, and are charged in accordance to their usage rates.

- **Aggregate**: It shows that cloud computing uses virtualization to consolidate multiple VMs into the same PMs. This characteristic is almost equivalent to '*multi-tenancy*' characteristic that is available in other publications that define cloud computing. The difference between the two is that aggregate include cases of private cloud, where VMs belonging to one user are consolidated within the same PMs.

- **Self-managing**: It shows capability of cloud computing to automatically reallocate VMs. This happens as a reaction to avoid situations that might lead to unwanted circumstances. Unwanted circumstances include, among others, overloading, faults and security breaches.

- **Shared resources**: This characteristic is almost equivalent to '*resource pooling*' available in other publications that define cloud computing. In addition to resource pooling characteristics that define seemingly unlimited resource pooling, this characteristic shows that consolidated VMs share physical resources such as memory, storage and CPU.

In addition to these newly defined characteristics are the discussed cloud computing characteristics which are common in currently existing publications. These are:

- **On-demand self-service**: It shows that cloud computing services are provisioned upon request and require **no** human interaction.

- **Broad network access**: It shows that cloud computing services are accessed remotely through the use of networks.

Using these characteristics, this study re-defined cloud computing and used the following definition:
"an on-demand self-service and self-managing computing utility that aggregates multiple instances of operating systems that share computing resources, and accessed through use of the internet"

10.2.2 Classification of VM Placement Algorithms

Literature review conducted in this study resulted in classification of existing VM placement algorithms. They were classified into at least one of the following categories: 1) VM placement algorithms that focus on QoS, 2) VM placement algorithms that focus on cost reduction, and 3) VM placement algorithms that focus on security.

- **VM placement algorithms that focus on QoS**: They are VM placement algorithms whose objectives strive to provide quality to cloud services. These objectives reduce possible performance degradations that are brought by *aggregate* and *shared resources* characteristics of the cloud. If not catered for, these characteristics may result in over-utilization of shared computing resources that results in performance degradations.

- **VM placement algorithms that focus on cost reduction**: The objectives of this kind of VM placement algorithms strive to reduce costs of administering cloud infrastructure. Most of these VM placement algorithms observe energy consumption minimization as a means of reducing costs. This is because, energy consumption has recently been a concern in cloud computing. It is caused by rapid increase in the use of energy in datacentres. This results in increasing costs of running these datacentres.

- **VM placement algorithms that focus on security**: These are VM placement algorithms that strive to reduce risks brought by cloud computing characteristics. Among others, aggregate and shared resources introduce cloud specific vulnerabilities that promote inter-VM attacks. There is therefore the need to strategically place VMs in a way that minimizes these attacks.

10.2.3 Hypervisor Benefits - VM Management Capabilities

In addition to available hypervisor benefits discussed in previous studies, the author in this study introduced VM management capabilities as the third benefit. This was introduced because the first two benefits, which are *isolation* and *resource sharing*, do not support all cloud computing characteristics. They only support aggregate and resource sharing characteristics. The other four characteristics of cloud computing are therefore supported by this newly introduced hypervisor benefit. It is important to note that hypervisors are virtualization software that supports provision of these cloud computing characteristics.

10.2.4 O-Sec VM Placement Algorithm Security Features

Of the two security features used in O-Sec VM placement algorithm, *isolation of critical from non-critical VMs* is unique for this study. It works towards minimization of inter-VM attacks brought by co-location of VMs with different vulnerability statuses. Failure to isolate these VMs results in malicious activists targeting and compromising vulnerable VMs. Once compromised, malicious users take advantage of shared resources to attack co-located VMs that are less vulnerable.

10.3 Publication Produced

Some of the results produced in this research were published in the following conference paper:
Motlatsi Isaac Thulo and J. H. P. Eloff. **Towards optimized security-aware (O-sec) VM placement algorithms.** *In Proceedings of the 3rd Inter-national Conference on Information Systems Security and Privacy Volume 1: ICISSP, pages 411-422, 2017.*

This paper presented an evaluation criteria that was used for selection of TPVMP. As mentioned, it used quantitative method to find the VM placement algorithm that assumed highest optimization and/or security objectives. It selected appropriate VM placement algorithm from a pool of publications searched from existing research databases.

10.4 Challenges Encountered

The main challenge encountered in this research was development of cloud computing environment for evaluation purposes. A number of options were tried in the development of this environment. However, some of these options failed due to unforeseen limitations. These options together with challenges encountered in their implementations are discussed in the next subsections.

10.4.1 Configuration of mini-cloud in the lab

The first option taken in implementation of the test environment was development of a mini-cloud infrastructure in the lab. The following configurations were used in developing this architecture:

- *Cloud management software:* Opennebula 4.12
- *Hypervisor:* KVM
- *Operating Systems:* Ubuntu server 16.04
- *Front-end:* HP ProLiant G5
- *End-notes:* HP ProLiant G4 and HP ProLiant G5

Problems:

1. Some of these workstations crashed while installing Opennebula. This occured because of the old workstations that were used in this experiment, which had reached their end-of-life (EOL).
2. Also, installation of more than one end-note could not be successful. This is also a result of workstations that had reached their EOL.

10.4.2 Use of CloudSim simulation environment

The second option in implementation of test environment was use of simulated cloud computing environment. The first simulation environment of choice was CloudSim. This was chosen because it is the most common cloud simulation environment used in experiments. In addition, it is the testing environment of choice in evaluations of TPVMP. The problems encountered include the following:

1. Coupled nature of CloudSim classes that limits its flexibility to re-define classes.

2. Duplicate code with limited documentation that restricts extensibility and maintainability of the tool.

Generally, CloudSim made it difficult to generate datacentre required for this study. This is because, it requires both implementation of new classes and modification of existing classes.

10.5 Future Work

This research was the first attempt towards notion of O-Sec VM placement algorithm. The aim was to provide an optimized and secure cloud environment through placement of VMs. Although the implemented VM placement algorithm achieved the goal of this research, there are some limitations that need to be addressed in future studies. These limitation include:

- The implemented algorithm is programmed to perform specific task only. It used a defined criteria to place VMs into available PMs. However, it lacks the concept of machine learning where VM placement can adapt to changing cloud architecture behaviour.

- This study proposed isolation of critical VM from non-critical VMs. However, it did not look into security of host PMs, more especially for those that host critical VMs. Ideally, these hosts need to be of higher security as compared to their counter partners.

It is therefore important to include concepts that address above mentioned limitations. That is, implementation of security-aware VM placement algorithms should include concepts of machine learning. They must adapt to changing datacentre conditions that might lead to compromise of the availability and integrity of placed VMs. This can also be achieved by integrating concepts of near-miss detection process in order to avoid unwanted datacentre conditions. Furthermore, future research need to enhance security to PMs, more especially those that host critical VMs. Some common security principles such as IDS, IPS and firewalls need to be taken into consideration in order to secure these PMs.

To successfully achieve the proposed concepts for future work, real infrastructure needs to be used. This is because it is difficult, if possible, to include PM security in simulated cloud infrastructures.

Bibliography

[1] Lynn S Aaron and Catherine M Roche. Teaching, learning, and collaborating in the cloud: Applications of cloud computing for educators in post-secondary institutions. *Journal of Educational Technology Systems*, 40(2):95–111, 2011.

[2] Zaina Afoulki, Aline Bousquet, and Jonathan Rouzaud-Cornabas. A security-aware scheduler for virtual machines on IaaS clouds. *Report 2011*, 2011.

[3] Saeed Al-Haj, Ehab Al-Shaer, and HariGovind V Ramasamy. Security-aware resource allocation in clouds. In *Services Computing (SCC), 2013 IEEE International Conference on*, pages 400–407. IEEE, 2013.

[4] Mazhar Ali, Samee U Khan, and Athanasios V Vasilakos. Security in cloud computing: Opportunities and challenges. *Information Sciences*, 305:357–383, 2015.

[5] Mauro Andreolini, Sara Casolari, Michele Colajanni, and Michele Messori. Dynamic load management of virtual machines in cloud architectures. In *International Conference on Cloud Computing*, pages 201–214. Springer, 2009.

[6] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[7] Anton Beloglazov and Rajkumar Buyya. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24(13):1397–1420, 2012.

[8] Sushil Bhardwaj, Leena Jain, and Sandeep Jain. Cloud computing: A study of Infrastructure as a Service (IaaS). *International Journal of engineering and information Technology*, 2(1):60–63, 2010.

[9] Ofer Biran, Antonio Corradi, Mario Fanelli, Luca Foschini, Alexander Nus, Danny Raz, and Ezra Silvera. A stable network-aware VM placement for cloud systems. In *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pages 498–506. IEEE Computer Society, 2012.

[10] T Bittman. Predicts 2004: Server virtualization evolves rapidly. *USA, Gartner Inc*, 2003.

[11] Norman Bobroff, Andrzej Kochut, and Kirk Beaty. Dynamic placement of virtual machines for managing SLA violations. In *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*, pages 119–128. IEEE, 2007.

[12] Sven Bugiel, Stefan Nürnberger, Thomas Pöppelmann, Ahmad-Reza Sadeghi, and Thomas Schneider. Amazonia: When elasticity snaps back. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS 11, page 389400, New York, NY, USA, 2011. Association for Computing Machinery.

[13] Rodrigo N Calheiros, Rajiv Ranjan, César AF De Rose, and Rajkumar Buyya. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services. *arXiv preprint arXiv:0903.2525*, 2009.

[14] Ricardo Stegh Camati, Alcides Calsavara, and Luiz Lima Jr. Solving the virtual machine placement problem as a multiple multidimensional knapsack problem. *ICN 2014*, page 264, 2014.

[15] Eddy Caron, Anh Dung Le, Arnaud Lefray, and Christian Toinard. Definition of security metrics for the cloud computing and security-aware virtual machine placement algorithms. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference on*, pages 125–131. IEEE, 2013.

[16] Mohammed Rashid Chowdhury, Mohammad Raihan Mahmud, and Rashedur M Rahman. Implementation and performance analysis of various VM placement strategies in cloudsim. *Journal of Cloud Computing*, 4(1):1, 2015.

[17] Peter M Corcoran. Cloud computing and consumer electronics: A perfect match or a hidden storm?[soapbox]. *IEEE Consumer Electronics Magazine*, 1(2):14–19, 2012.

[18] Jan HP Eloff. Selection process for security packages. *Computers & Security*, 2(3):256–260, 1983.

[19] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08*, pages 1–10. Ieee, 2008.

[20] Mauro Gaggero and Luca Caviglione. Model predictive control for energy-efficient, quality-aware, and secure virtual machine placement. *IEEE Trans. Automation Science and Engineering*, 16(1):420–432, 2019.

[21] Manish Godse and Shrikant Mulik. An approach for selecting Software-as-a-Service (SaaS) product. In *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on*, pages 155–158. IEEE, 2009.

[22] Vânia Gonçalves and Pieter Ballon. Adding value to the network: Mobile operators experiments with Software-as-a-Service and Platform-as-a-Service models. *Telematics and Informatics*, 28(1):12–21, 2011.

[23] Nelson Gonzalez, Charles Miers, Fernando Redigolo, Marcos Simplicio, Tereza Carvalho, Mats Näslund, and Makan Pourzandi. A quantitative analysis of current security concerns and solutions for cloud computing. *Journal of Cloud Computing: Advances, Systems and Applications*, 1(1):11, 2012.

[24] Robert Grossman, Yunhong Gu, Michal Sabala, Collin Bennet, Jonathan Seidman, and Joe Mambratti. The open cloud testbed: A wide area testbed for cloud computing utilizing high performance network services. *arXiv preprint arXiv:0907.4810*, 2009.

[25] Yi Han, Jeffrey Chan, Tansu Alpcan, and Christopher Leckie. Virtual machine allocation policies against co-resident attacks in cloud computing. In *Communications (ICC), 2014 IEEE International Conference on*, pages 786–792. IEEE, 2014.

[26] Yi Han, Jeffrey Chan, Tansu Alpcan, and Christopher Leckie. Using virtual machine allocation policies to defend against co-resident attacks in cloud computing. *IEEE Transactions on Dependable and Secure Computing*, 14(1):95–108, 2017.

[27] Keiko Hashizume, David G Rosado, Eduardo Fernández-Medina, and Eduardo B Fernandez. An analysis of security issues for cloud computing. *Journal of Internet Services and Applications*, 4(1):5, 2013.

[28] Brian Hayes. Cloud computing. *Commun. ACM*, 51(7):9–11, July 2008.

[29] Zongjian He and Guanqing Liang. Research and evaluation of network virtualization in cloud computing environment. In *Networking and Distributed Computing (ICNDC), 2012 Third International Conference on*, pages 40–44. IEEE, 2012.

[30] Ines Houidi and Djamal Zeghlache. Exact adaptive virtual network embedding in cloud environments. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013 IEEE 22nd International Workshop on*, pages 319–323. IEEE, 2013.

[31] Motlatsi i Isaac Thulo and J. H. P. Eloff. Towards Optimized Security-aware (O-sec) VM placement algorithms. In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,*, pages 411–422, 2017.

[32] Andrzej Kochut and Kirk Beaty. On strategies for dynamic resource management in virtualized server environments. In *2007 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 193–200. IEEE, 2007.

[33] Miika Komu, Mohit Sethi, Ramasivakarthik Mallavarapu, Heikki Oirola, Rasib Hassan Khan, and Sasu Tarkoma. Secure networking for virtual machines in the cloud. In *CLUSTER Workshops*, pages 88–96. Citeseer, 2012.

[34] Chunguang Kuang, Qing Miao, and Hua Chen. Analysis of software vulnerability. *WSEAS Transactions on Computers Research*, 1(1):45, 2006.

[35] Chin-Fu Kuo, Ting-Hsi Yeh, Yung-Feng Lu, and Bao-Rong Chang. Efficient allocation algorithm for virtual machines in cloud computing systems. In *Proceedings of the ASE BigData & SocialInformatics 2015*, page 48. ACM, 2015.

[36] Butler W Lampson. A note on the confinement problem. *Communications of the ACM*, 16(10):613–615, 1973.

[37] Kangkang Li, Huanyang Zheng, Jie Wu, and Xiaojiang Du. Virtual machine placement in cloud systems through migration process. *International Journal of Parallel, Emergent and Distributed Systems*, 30(5):393–410, 2015.

[38] Ching-Chi Lin, Pangfeng Liu, and Jan-Jan Wu. Energy-efficient virtual machine provision algorithms for cloud systems. In *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, pages 81–88. IEEE, 2011.

[39] Liang Liu, Hao Wang, Xue Liu, Xing Jin, Wen Bo He, Qing Bo Wang, and Ying Chen. Greencloud: a new architecture for green data center. In *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, pages 29–38. ACM, 2009.

[40] Fabio Lopez-Pires and Benjamin Baran. Virtual machine placement literature review. *arXiv preprint arXiv:1506.01509*, 2015.

[41] Zoltán Ádám Mann and Máté Szabó. Which is the best algorithm for virtual machine placement optimization? *Concurrency and Computation: Practice and Experience*, 29(10), 2017.

[42] Mohammad Masdari, Sayyid Shahab Nabavi, and Vafa Ahmadi. An overview of virtual machine placement schemes in cloud computing. *Journal of Network and Computer Applications*, 66:106–127, 2016.

[43] Peter Mell and Tim Grance. The NIST definition of cloud computing. *National Institute of Standards and Technology*, 53(6):50, 2009.

[44] Xiaoqiao Meng, Vasileios Pappas, and Li Zhang. Improving the scalability of data center networks with traffic-aware virtual machine placement. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[45] Dave Mitchell. Defining Platform-as-a-Service, or PaaS. *Bungee Connect Developer Network*, 2008.

[46] Bhavani B Nagesh et al. Resource provisioning techniques in cloud computing environment-a survey. *IJRCCT*, 3(3):395–401, 2014.

[47] NIST. Cve-2017-14319 detail, September 2017.

[48] Satoru Ohta. Virtual machine placement algorithms to minimize physical machine count. In *APNOMS*, pages 1–3, 2013.

[49] Diego Perez-Botero, Jakub Szefer, and Ruby B Lee. Characterizing hypervisor vulnerabilities in cloud computing servers. In *Proceedings of the 2013 international workshop on Security in cloud computing*, pages 3–10. ACM, 2013.

[50] Charles P Pfleeger and Shari Lawrence Pfleeger. *Analyzing computer security: a threat/vulnerability/countermeasure approach*. Prentice Hall Professional, 2012.

[51] Radu Prodan and Simon Ostermann. A survey and taxonomy of infrastructure as a service and web hosting cloud providers. In *Grid Computing, 2009 10th IEEE/ACM International Conference on*, pages 17–25. IEEE, 2009.

[52] Ebrahim Randeree, Rajiv Kishore, and H Raghav Rao. Investigating trust in outsourcing. *Eric K. Clemons University of Pennsylvania Thomas H. Davenport Accenture Institute for Strategic Change and*, page 135, 2008.

[53] Jenni Susan Reuben. A survey on virtual machine security. *Helsinki University of Technology*, 2:36, 2007.

[54] KA Scarfone. *Guide to security for full virtualization technologies*, volume 800. DIANE Publishing, 2011.

[55] Ben Shneiderman. Response time and display rate in human performance with computers. *ACM Computing Surveys (CSUR)*, 16(3):265–285, 1984.

[56] Manoel C Silva Filho, Raysa L Oliveira, Claudio C Monteiro, Pedro RM Inácio, and Mário M Freire. Cloudsim plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*, pages 400–406. IEEE, 2017.

[57] Utkal Sinha and Mayank Shekhar. Comparison of various cloud simulation tools available in cloud computing. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(3), 2015.

[58] Avvari Sirisha and G Geetha Kumari. API access control in cloud using the role based access control model. In *Trendz in Information Sciences & Computing (TISC), 2010*, pages 135–137. IEEE, 2010.

[59] Dinkar Sitaram, Phalachandra Hallymysore, Sudheendra Harwalkar, Supriya Kathare, Anirudh Srinivasan, Archana Reddy, Konark Kumar, and Sahana Sekhar. Opensim: A simulator of openstack services. In *Modelling Symposium (AMS), 2014 8th Asia*, pages 90–96. IEEE, 2014.

[60] Stephen Soltesz, Herbert Pötzl, Marc E Fiuczynski, Andy Bavier, and Larry Peterson. Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In *ACM SIGOPS Operating Systems Review*, volume 41, pages 275–287. ACM, 2007.

[61] Ilango Sriram. Speci, a simulation tool exploring cloud-scale data centres. *Cloud Computing*, pages 381–392, 2009.

[62] YEVGENIY Sverdlik. Heres how much energy all us data centers consume. *DataCenterKnolwedge. com. June*, 27, 2016.

[63] T Swathi, K Srikanth, and S Raghunath Reddy. Virtualization in cloud computing. *International Journal of Computer Science and Mobile Computing, ISSN*, pages 540–546, 2014.

[64] Jakub Szefer, Eric Keller, Ruby B Lee, and Jennifer Rexford. Eliminating the hypervisor attack surface for a more secure cloud. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 401–412. ACM, 2011.

[65] Jakub Szefer and Ruby B Lee. A case for hardware protection of guest VMs from compromised hypervisors in cloud computing. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, pages 248–252. IEEE, 2011.

[66] S Venkata, Krishna Kumar, and S Padmapriya. A survey on cloud computing security threats and vulnerabilities. *International Journal of Innovation Research in Electrical, Electronics, Instrumentation and Control Engineering*, Vol. 2(Issue 1):622–625, January 2004.

[67] Hieu Trong Vu and Soonwook Hwang. A traffic and power-aware algorithm for virtual machine placement in cloud data center. *International Journal of Grid & Distributed Computing*, 7(1):350–355, 2014.

[68] Jiang Wang, Angelos Stavrou, and Anup Ghosh. Hypercheck: A hardware-assisted integrity monitor. In *Recent Advances in Intrusion Detection*, pages 158–177. Springer, 2010.

[69] Bhathiya Wickremasinghe, Rodrigo N Calheiros, and Rajkumar Buyya. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 446–452. IEEE, 2010.

[70] Yuchen Wong and Qingni Shen. Secure virtual machine placement and load balancing algorithms with high efficiency. In *IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications, ISPA/IUCC/BDCloud/SocialCom/SustainCom 2018, Melbourne, Australia, December 11-13, 2018*, pages 613–620, 2018.

[71] Chiachih Wu, Zhi Wang, and Xuxian Jiang. Taming hosted hypervisors with (mostly) deprivileged execution. In *NDSS*, 2013.

[72] Hanqian Wu, Yi Ding, Chuck Winer, and Li Yao. Network security for virtual machine in cloud computing. In *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*, pages 18–21. IEEE, 2010.

[73] Mingdi Xin and Natalia Levina. Software-as-a-service model: Elaborating client-side adoption factors. 2008.

[74] Jing Xu and José Fortes. A multi-objective approach to virtual machine management in datacenters. In *Proceedings of the 8th ACM international conference on Autonomic computing*, pages 225–234. ACM, 2011.

[75] Jyun-Shiung Yang, Pangfeng Liu, and Jan-Jan Wu. Workload characteristics-aware virtual machine consolidation algorithms. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 42–49. IEEE, 2012.

[76] Xuebiao Yuchi and Sachin Shetty. Enabling security-aware virtual machine placement in IaaS clouds. In *Military Communications Conference, MILCOM 2015-2015 IEEE*, pages 1554–1559. IEEE, 2015.

[77] Amal Zaouch and Faouzia Benabbou. Load balancing for improved quality of service in the cloud. *International Journal of Advanced Computer Science and Applications IJACSA*, 6(7):184–189, 2015.

[78] Dimitrios Zissis and Dimitrios Lekkas. Addressing cloud computing security issues. *Future Generation computer systems*, 28(3):583–592, 2012.