Policy-Based Management Approach for Energy Efficient Cloud Computing Infrastructure

Fadi Alhaddadin

A thesis submitted to
Auckland University of Technology
in partial fulfillment of the requirements for the degree
of
Master of Computer and Information Sciences (MCIS)

2014

School of Computer and Mathematical Sciences

To my father and mother

إلى الأحباء أبي وأمي

Table of Contents

List of Figures

List of Tables

Attestation of Authorship

“I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.”

Signature of Candidate:

……

Dedication

The education of me was of paramount importance to my parents, and in addition to their strong encouragement, they were prepared to make any sacrifice to further my intellectual development.

I owe a deep debt of gratitude to my lovely father and lovely mother. Their support, encouragement, and constant love have sustained me throughout my life. I am lovingly dedicating this thesis to my parents, a special feeling of gratitude to both of them.

Dear Dad and Mom, I cannot thank you enough for all the support and love you have given me. I never would have made it here without you. Thank you for everything!

Acknowledgements

It would not have been possible to write this master thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

Foremost, I would like to express my deep and sincere gratitude to my supervisor, Dr. William Liu, of the School of Computer and Mathematical Sciences, Auckland University of Technology. His wide knowledge and logical way of thinking have been of great value for me. His understanding, encouragement and personal guidance have provided a good basis for the present thesis. I would also like to express my deep gratefulness to my co-supervisor, Dr. Jairo A. Gutiérrez for his great help and guidance whenever it was needed. His pain-staking effort in proof reading the drafts is greatly appreciated.

I owe my loving thanks and appreciation to my parents, to whom I am dedicating this thesis. Without their continuous encouragement, support and understanding it would have been impossible for me to finish this work.

A sincere thank is given to Network and Security Research Group (NSRG) for their valuable suggestions and feedback to my research and conference presentations. The facilities provided and the financial support of the contestable fund provided by School of Computer and Mathematical Sciences Research Committee is gratefully acknowledged. This generous support enabled me to attend the NZCSRSC'13 conference which was held

in Hamilton in April 2013 to present my accepted paper entitled: A Policy-Based Management Approach for Greening the Cloud Infrastructure.

Last but not least, a sincere thank is given to Shoba Tegginmath, the postgraduate CIS programme leader for her prompt responses and advices whenever it was needed throughout my research.

Publications and Presentations

Conference Paper

Fadi Alhaddadin, W. Liu, J. A. Gutiérrez, “A Policy Based Management Approach for Greening the Cloud Infrastructure”, New Zealand Computer Science Research Student Conference (NZCSRSC), Hamilton, New Zealand, April 2013.

Presentations

1. New Zealand Computer Science Research Student Conference (NZCSRSC). Topic: A Policy Based Approach for Greening the Cloud Infrastructure. Hamilton, New Zealand, 2013
2. Network and Security Research Group (NSRG) weekly meeting, research proposal July 2012
3. Network and Security Research Group (NSRG) weekly meeting, report on 3 months literature reviews, October 2012
4. Network and Security Research Group (NSRG) weekly meeting, progress report presentation, March 2013.

Abstract

Cloud computing technology is gaining great popularity in our day due to the utility-oriented information technology services that it offers worldwide. Due to the pay-as-you-go elasticity that cloud computing technology facilitates, hosting pervasive applications has become possible from various ends such as consumer, scientific, and business domains. The technology of Cloud Computing facilitates a computing-as-a-service model where computing resources are made available as a utility service. However, although cloud computing technology returns great benefits in different aspects, data centers consume significant amounts of electricity in order to run, hence they require high operational costs and cause harmful outcomes to the environment such as carbon footprints and emissions. Therefore, we found it valuable to design a new cloud system model which contributes towards reducing the energy consumption of cloud datacenters, with consideration to the quality of service delivered.

In this thesis, we have proposed a new cloud system model and architecture that is able to handle and manage a group of virtual machine migration heuristic algorithms proposed by other researchers. The proposed system model, User Profile Aware Policy Switching algorithm (UPAPS) framework has proven ability in managing a group of these heuristic algorithms by employing our proposed architectural components namely: the UPAPS algorithm and the User Service Profile (USP). The UPAPS algorithm together with the USP component contribute to the efficient management of heuristic algorithms and therefore have achieved the desired trade-off between energy consumption and quality of

service delivered. The USP component in the UPAPS framework is intended to be the instructive part in the system model which is configured according to users' requirements. It contains the requirements of the user for other components of the system to work accordingly. The UPAPS component is part of the policy-based management and it functions according to the instruction held in the USP component.

The extensive simulation results have shown a great improvement on the cloud infrastructure in terms of power efficiency and resource management using our proposed approach (i.e., UPAPS framework). The validation of our proposed system model has been done through conducting two different cloud scenario studies under both the current system model and the proposed system model. Simulation results for both systems were compared, and the UPAPS framework showed significant reduction in energy consumption in comparison to the current system without violating the quality of service required by the cloud users in both scenarios.

Chapter 1: INTRODUCTION

Cloud computing technology represents a different method for architecting and remotely managing computer resources [1]. The term “Cloud Computing” is a new name that refers to an old concept; the delivery of computing services from a remote location, analogous to the way electricity, water, and other utilities are provided to most customers [2]. Cloud computing services are delivered through a network which is usually the internet. Cloud computing is a model that enables ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released without a need for great management efforts or service provider interaction [3].

The technology of Cloud Computing facilitates a computing-as-a-service model where computing resources are made available as a utility service. It allows the availability of as much resources as demanded by the user in a pay-as-you-go basis, which makes it different from the earlier computing models in which enterprises have to invest enormous funds to implement and build their own IT infrastructures [4]. It facilitates the possibility for users to rent only at the time of need the desired amount of computing resources out of a huge mass of distributed computing resources without worrying about the locations or internal structures of these resources [5].

Cloud computing leverages the virtualization of computing resources aiming at allowing customers to provision resources on-demand [6]. Today, cloud computing technology is regarded as an important trend towards the future’s distributed and ubiquitous computing services offered over the global internet [7]. It is also gaining a great deal of attention and

popularity in our current society due to the benefits that it can flexibly offer to its users with various applications for various purposes within the context of a pay-as-you-go model.

1.1 Background

Despite the great benefits that cloud computing technology returns, the spread of cloud computing datacenters has led to establishing huge-scale data centers which comprise of thousands of computing nodes and consume massive amounts of electrical energy [8]. Power consumption is considered one of the main issues related to the success of cloud computing. Due to the ever growing number of large cloud computing datacenters, the electrical energy consumed by hardware facilities and cooling systems has increased notably.

Greenpeace estimated that the total cloud energy consumption (datacenters plus telecommunications) would be up to 622.6 billion Kwh (Kilo Watt per Hour) in 2007 [9]. A research conducted in 2010 showed that cloud computing data centers in 2010 were responsible for 1.1% to 1.5% of the total power consumed in the world [10]. As a consequence of the huge amount of energy consumed by cloud computing datacenters, Gartner in 2007 estimated that the information and communication technologies (ICT) industry generates approximately 2% of the total global CO^2 emissions which is equal to the aviation industry [11]. Surprisingly, the main reason behind the huge amount of power consumption that cloud computing is responsible for is not only the quantity of computing resources or the inefficiency in power consumption of the hardware devices, but rather lies in the inefficient usage of these resources [8].

One of the major causes of energy inefficiency in datacenters is the idle power wasted when servers run at low utilization. Even at a very low CPU load/utilization, the power consumed is over 50% of the peak power [12]. A research involved collecting data from more than 5000 production servers in a period of six months has shown that although servers usually are not idle, their utilization rarely approaches 100% as most of the time they operate at 10-50% of their full capacity; this leads to extra expenses on over-provisioning and thus extra total cost of acquisition (TCA) [13]. Thus, keeping servers underutilized contributes to a great inefficiency from the prospective of power consumption, which is today considered as a critical problem in the field of cloud computing.

1.2 Motivation

The objective of this thesis is to study the energy consumption issue in cloud-based environments. It focuses on the utilization management of computing resources by using the virtualization approach. The main goal is to reduce the amount of energy consumed by a datacenter during its idle status time and/or during the time in which the utilization of its resources is not worth keeping it running and consuming energy at the maximum capacity.

Virtualization allows cloud providers to create multiple Virtual Machines (VMs) instances on a single physical server for the goal of improving the utilization of resources [8]. Virtualization technology also contributes towards eliminating the power consumption by switching idle nodes to low-power modes such as sleep mode or hibernation using live virtual machine migration [14]. The use of virtual machine migration technology also allows dynamic consolidation of virtual machines. Virtual machine consolidation allows

reducing the number of running physical nodes according to their resource requirements as part of resource management.

However, the accuracy and efficiency of cloud management are very important, because aggressive consolidation of virtual machines can lead to performance degradation, and therefore violation to the service level agreement (SLA) established between the cloud providers and their customers [8].

1.3 Research Objectives

This thesis focuses on tackling the issue of energy consumption in cloud computing datacenters in the current cloud system model and architecture. This thesis aims at developing and proposing a new management framework for cloud-based environments which achieves sufficient trade-offs between energy consumption and Quality of Service, and therefore overcoming the issue of energy insufficiency in the cloud-based environment by proposing a new cloud architecture with consideration to the management of the computing resources and quality of service delivered at the same time. We aim at contributing towards more means of elasticity in the field of cloud computing. Overall, there are two major objectives this research aims at in the field:-

Firstly, we intend to propose a management framework that achieves a sufficient trade-off between energy consumption in datacenters without aggressively affecting the Quality of Service delivered. Several virtual machine migration algorithms (Heuristic algorithms) have been developed and proposed by other researchers for the use of virtual machine migrations in virtualized cloud environments [8] however, none of these algorithms can

satisfy the need of the desired trade-off between energy efficiency and Quality of Service (QoS) on its own. The UPAPS framework intends to prove its ability in handling and managing a group of adaptive heuristic algorithms in virtualized cloud-based environment, with the goal of addressing the issue of underutilization of computing resources in the current cloud system design which causes inefficiency in energy consumption in cloud datacenters.

The second objective of this research is to propose a new cloud-based architecture concept that offers a range of Service Level Agreements (SLA) via a policy-based management framework.

With our proposed management framework, users are more involved in the management of their own profiles and the system facilitates more flexibility in terms of service schemes that a cloud provider can offer to users.

The main goal of the proposed architecture is the reduction of energy consumption on the cloud provider side. This eliminates the operational costs of the cloud hardware and therefore reduces the cost of the service at the cloud user side. The current implementation of the cloud computing system model is not energy efficient due to the underutilization of computing resources. The underutilization of computing resources is not limited to extra energy consumption and hence operating costs in the cloud provider side, but also causes cloud clients to pay for what they do not really need. For example, clients may demand the highest available QoS from the cloud provider for a certain time window during the day, while they do not need the same QoS at a different time window. According to the current system model, for clients, in order to satisfy their requirements during the time window in

which the highest available QoS is needed, they need to purchase the highest available quality of service throughout the day hence, they are currently in a position to pay for what they do not really demand for their individual's/organization's needs in one full day. In this thesis, the proposed cloud architecture aims at allowing cloud clients to manage their service schemes according to their actual needs in any particular time window during any period of time.

1.4 Thesis Structure

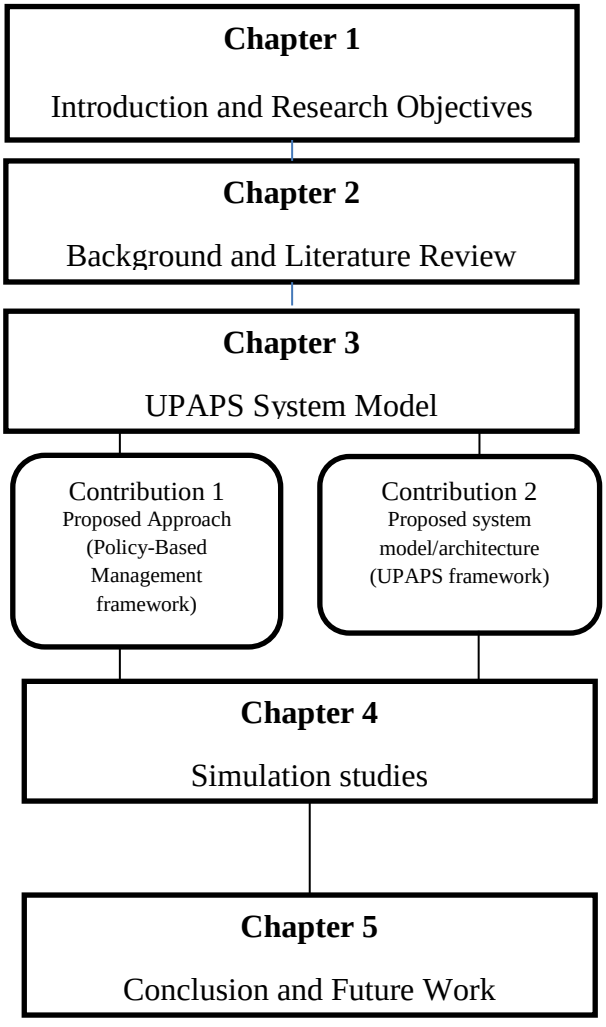


FIGURE 1.1 THESIS STRUCTURE

The thesis's structure is illustrated in figure 1.1. The remainder of this thesis contains the introduction of relevant background knowledge, followed by a description and comprehensive discussion of cloud computing technology. The extensive simulation studies are then presented to validate the proposed approach and system model. The chapters of the thesis are organized as the following:

Chapter 2 presents a background introduction to the technology of cloud computing, it briefly explains the technology of cloud computing and discusses its service types, characteristics and deployment models. Moreover, it introduces the key issue of the inefficiency in energy consumption in cloud computing and the root causes of it. Furthermore, an overview of the Policy-Based Network Management (PBNM) approach towards better management of cloud networks is presented based on the existing literature, with a focus on how it can be used to overcome the issue of inefficiency in energy consumption in cloud configurations.

Chapter 3 presents the proposed approach which aims at overcoming the issue of underutilization of computing resources and therefore inefficient energy consumption datacenters. It identifies the dynamic components and metrics involved in the approach and explains how they work towards the objective of the approach. Moreover, it outlines the issues with the current cloud system model and likewise, it draws the proposed cloud model (UPAPS) model and its management framework with detailed justification on how they enhance the field of cloud computing towards tackling the issues encountered with the current cloud system model.

Chapter 4 presents extensive simulation studies and discussion on the proposed system architecture. Simulation processes have been conducted on both the current system model and the UPAPS framework. Results showed that the proposed system architecture performs better than the current system in terms of power efficiency. Moreover, the UPAPS framework has been validated in terms of providing more elasticity for users in choosing their ICT service schemes which makes it more cost efficient in comparison to the current system model of cloud computing.

Finally, contributions and findings are summarized in **Chapter 5**. In addition, we describe the limitations of this research and suggest some possible research directions to advance our proposed approach/system model.

CHAPTER 2: BACKGROUND

In this chapter, an over view of cloud computing technology and its typical underlying model and architecture is provided. The main concern of energy consumption in the same field is also introduced and its main reasons are pointed out and described. Furthermore, we also present the current literature review and related work that has been conducted on the issue of energy consumption in cloud computing towards overcoming it.

This chapter is organized as follows: Section 2.1 presents an overview of cloud computing technology and its basic service layers, characteristics and deployment models. Section 2.2 presents the methodology followed for this thesis. Section 2.3 introduces the issue of inefficient energy consumption in cloud computing with a brief explanation about the root causes of it. Section 2.4 presents Policy-Based Network Management (PBNM) which has been proposed by others for the goal of improving the management of computer networks. PBNM is presented as a potential solution towards overcoming the issue of underutilization of computing resources which therefore contributes towards solving the problem of inefficiency in energy consumption in the field of cloud computing. Finally, section 2.5 summarizes the chapter.

2.1 Overview of Cloud Computing

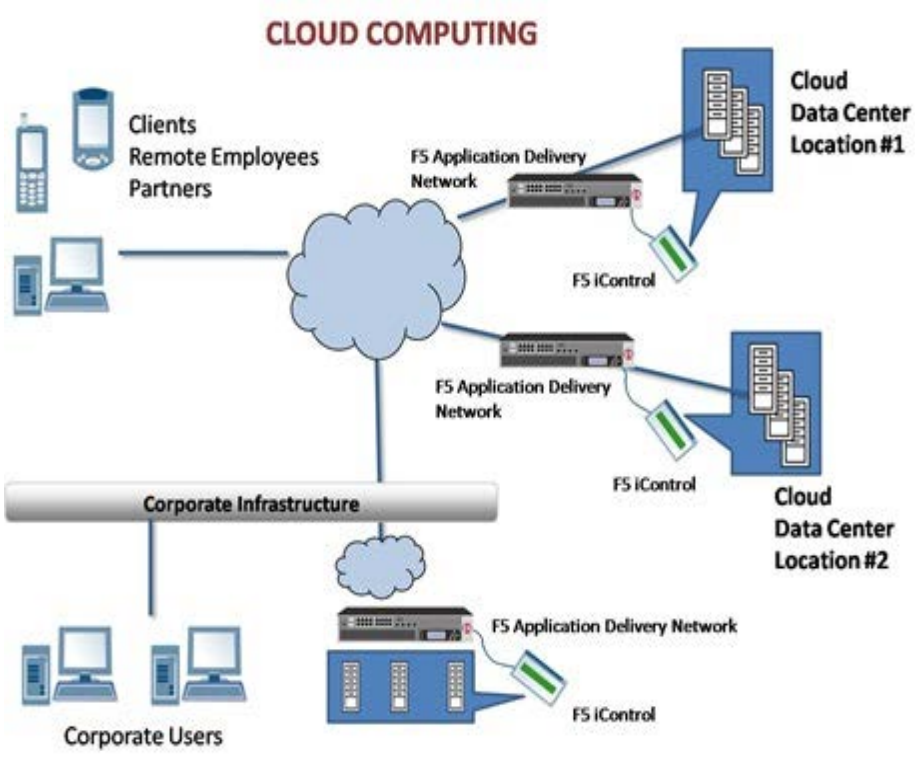


FIGURE 2.1 CLOUD COMPUTING OVERVIEW [15]

Cloud computing can be defined as a style of computing where massively scalable IT-related functions and information are provided as a service across the Internet, potentially to multiple external customers, where the consumers of the services need only care about what the service does for them, not how it is implemented. Figure 2.1 presents an overview of cloud computing technology that demonstrates its scalable architecture that can satisfy various types of users [15]. Cloud computing is an alternative delivery and acquisition model for IT-related services [16]. Cloud computing can be defined as a model for enabling convenient, on-demand network access to a shared pool of configurable resources such as

networks, storage, applications and services which can be rapidly provisioned and released with minimal management efforts or service provider interaction [17]. It is a paradigm that focuses on sharing data and computation over a scalable network of nodes with the main goal of using the existing infrastructure to bring all feasible services to the cloud and to make them accessible at any point of time and location [18].

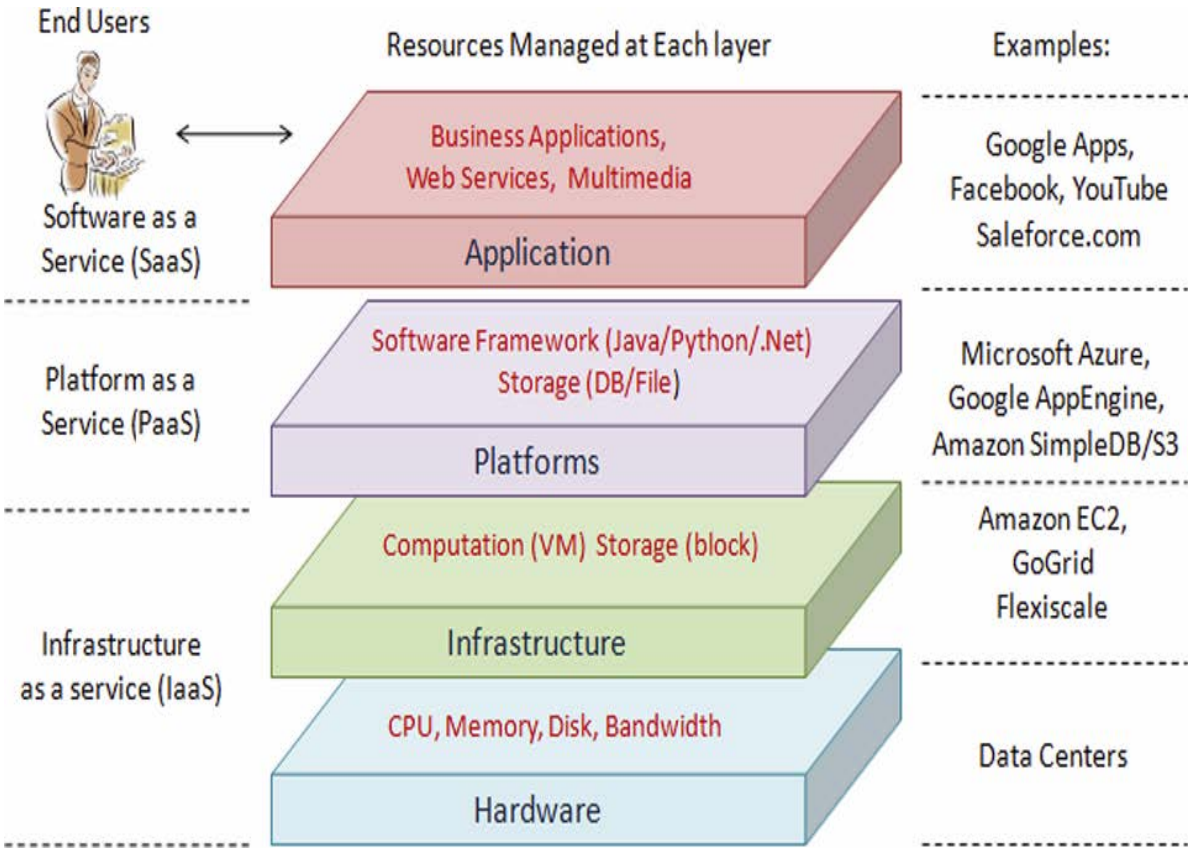


FIGURE 2.2 CLOUD COMPUTING SERVICE MODELS [30]

2.1.1 Cloud computing service models

Cloud computing technology offers services in three primary models; Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [19]. These

service models represent the primary cloud services provided by cloud providers according to users' requirements. Cloud computing technology can be split into three models in a pyramidal shape as shown in figure 2.2.

Software as a Service (SaaS) is an application delivery model that allows users to utilize a software solution over the internet [3] . It is a model that allows accessing an application that is hosted in a remote datacenter via internet connection. In SaaS, users are able to purchase accessibility and usability of an application or service that is hosted in the cloud. In the SaaS service model, applications are accessible from various devices through user interfaces. Users are not able to manage or control the underlying cloud infrastructure such as networks, servers and operating systems however; there is a possible exception of limited user-specific application configuration settings [3]. Facebook is an example of a SaaS service; Facebook users can create and access their social/commercial accounts on the Facebook site through any internet enabled devices while the service is hosted in a remote datacenter.

Platform as a Service (PaaS) is the service that allows cloud users to access platforms and deploy their own software/applications onto the cloud infrastructure. PaaS allows the creation of web applications quickly and easily and without the complexity of buying and maintaining the software infrastructure underneath it [20]. In PaaS, users create their own software using tools and libraries provided by cloud providers, but they are not permitted to manage or control the underlying cloud infrastructure such as network, servers, operating systems...etc., however, they are able to control their deployed applications and have access to the configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS) is the service model of cloud computing that allows users to manage and control operating systems, applications, and storage and network connectivity, without controlling the cloud infrastructure [19]. IaaS is defined as a way of delivering cloud computing infrastructure such as servers, storage, network and operating systems [20]. In IaaS, computing resources are distributed as a measured, scalable service with variable costs and pricing model. The main benefit of IaaS is that it helps users avoid the expense and need for dedicated systems for their organization and/or their associated staffing needs. It generally includes numerous users sharing the capabilities of a single piece of hardware.

2.1.2 Cloud computing characteristics

The National Institute of Standards and Technology (NIST) points out 5 major characteristics in cloud computing; On-demand self-service, broad network access, Resource pooling, rapid elasticity and measured service [3].

On-demand self-service allows cloud users to individually provision computing capabilities as needed without a need for human interaction with each service provider.

Broad network access allows the availability of capabilities over the network which can be accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms such as mobile devices.

Resource pooling characteristic allows multiple users to be served using a multi-tenant model. In resource pooling, users are serviced with different physical and virtual resources

which are dynamically assigned and reassigned according to users' requirements. Those resources may include memory, storage and bandwidth.

Rapid elasticity refers to the capability of delivering services at any time and quantity according to users' requirements.

Measured service allows the feasibility of measuring and controlling the computing resources usage by leveraging a metering capability at some level of abstraction appropriate to the type of service such as storage and processing.

2.1.3 Cloud computing deployment models

In order to obtain the benefits that cloud computing technology offers for individuals and/or organizations, it is very essential for cloud clients to identify the requirements that need to be addressed by the cloud. In [3], there are four cloud deployment models recommended by the National Institute of Standards and Technology (NIST); Private cloud, Community cloud, Public cloud, and Hybrid cloud. Figure 2.3 illustrates the four mentioned cloud models in which each of them serves clients according to their requirements.

Private cloud is the model of cloud computing that is provisioned exclusively for a single user which is usually an organization. Private cloud may be owned, operated and managed by its client. Private clouds can be hosted internally or externally. One of the main objectives of implementing a private cloud is to avoid the security issues as it is implemented safely with private firewall and other security means.

Public cloud is the cloud model that is provisioned for public use. The reason of its name "public" is because it is meant to be accessed by users from the public. Public clouds are owned by various types of organizations, such as business organizations, academic organizations, government organizations or combinations of them. Public clouds are hosted on the premises of the cloud provider. For example, government organization hosts their pubic clouds in their own building/premises.

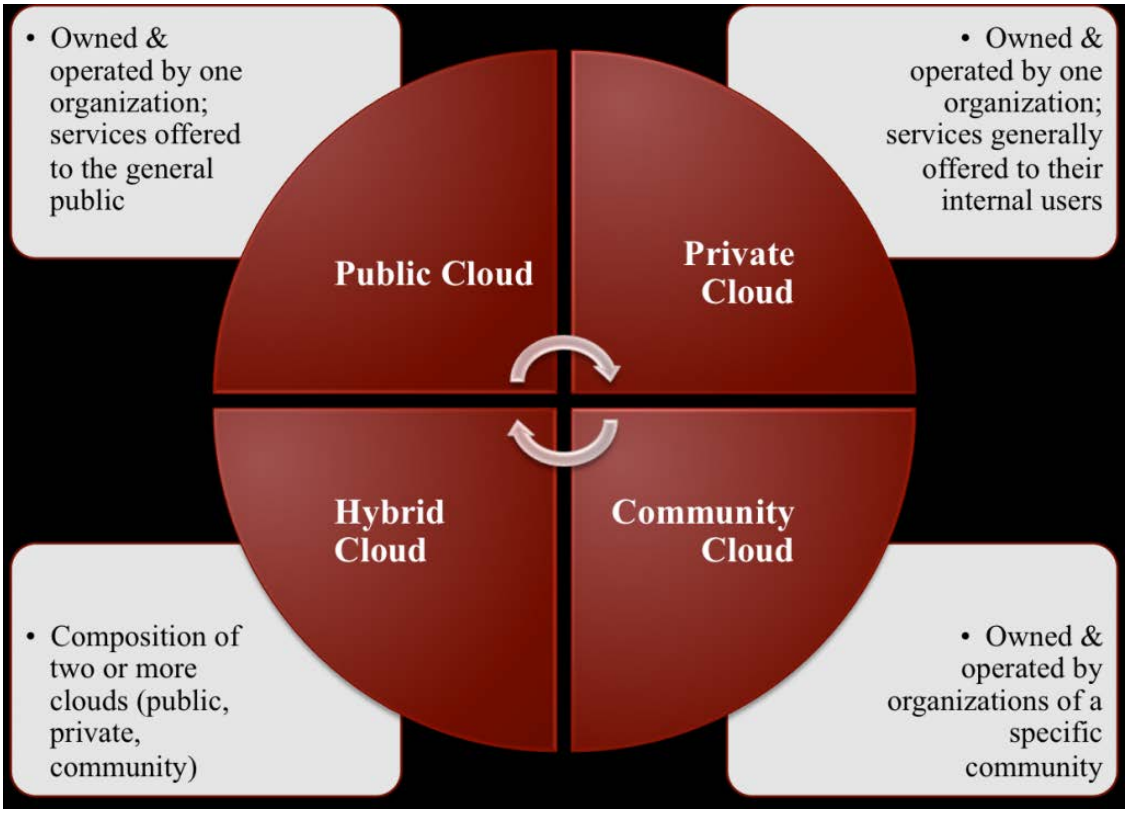


FIGURE 2.3 CLOUD DEVELOPMENT MODELS [21]

Hybrid cloud is considered as a composition of two or more distinct cloud implementations mentioned earlier, (private, community, or public). Hybrid clouds are unique entities, bound together by standardized technology that allows for data and application portability and interoperability. An example of hybrid cloud can be the cloud

bursting for load balancing between clouds. Hybrid clouds can be hosted internally and externally.

Community cloud model is provisioned to exclusively serve a specific community of users who may have similar concerns such as missions, policy and compliance consideration. The Community cloud may be owned, operated and managed by one or more of the organizations within a community. Community clouds can be hosted internally or externally.

2.2 Research Methodology

In order to achieve the objectives of this research, we opt to employ the modeling and simulation methodology described in [22]. Due to the nature of the study, where a new system model is being constructed and proposed, we find it useful to employ the modeling and simulation methodology, because such methodology enables us to describe the proposed model and further validate it through simulation processes where it becomes possible to compare results obtained by simulating both models, current and proposed models.

In [23], model validation is usually defined to mean “substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model. In [24], the validation term refers to two main aspects: conceptual validation and results validation.

Conceptual validation is when the anticipated reliability of the model or simulation conceptual model is assessed. The validity of the conceptual model determines that the theories and assumptions underlying the conceptual model are accurate and the model's representation of the problem entity and the model's structure, logic, and mathematical and causal relationships are reasonable for the intended purpose of the model.

Results validation is when results from the simulation of the implemented model are compared with an appropriate referent to demonstrate that the model can in fact support the intended use.

Modeling is defined as the process of producing a model; a model is a representation of the construction and working of some system of interest [25]. Simulation is the process of checking the operation of a particular model. For this research, due to the fact that it is too expensive to reconfigure an actual cloud system model and experiment with it for the sake of learning, we find it useful to use the simulation methodology on our proposed system model for validating it and proving its efficiency.

Simulation is a tool to evaluate the performance of a system, existing or proposed under different configurations of interest and over long periods of real time [25]. The simulation tool used for this research is the CloudSim toolkit [26] which is described further in this thesis. The CloudSim toolkit provides the facility of running cloud computing scenarios on certain (configurable) network configurations for chosen a period of time, with the goal of learning the behavior of the simulated system model under many conditions which include different energy consumption rates and levels of quality of service delivered.

Due to the nature of this research which involves designing a new cloud-based system model and validating it towards overcoming the issue of energy consumption in cloud computing, we have opted to employ Hevner's research cycle [27] which is shown in figure 2.4.

As seen in figure 2.4, the top half of the research cycle refers to the descriptive part of the research while the bottom half represents the prescriptive part of it which involves the corresponding activities of the research.

The research cycle points to two starting points for the research depending on the research objectives to be addressed. For our research, the starting point was the descriptive part,

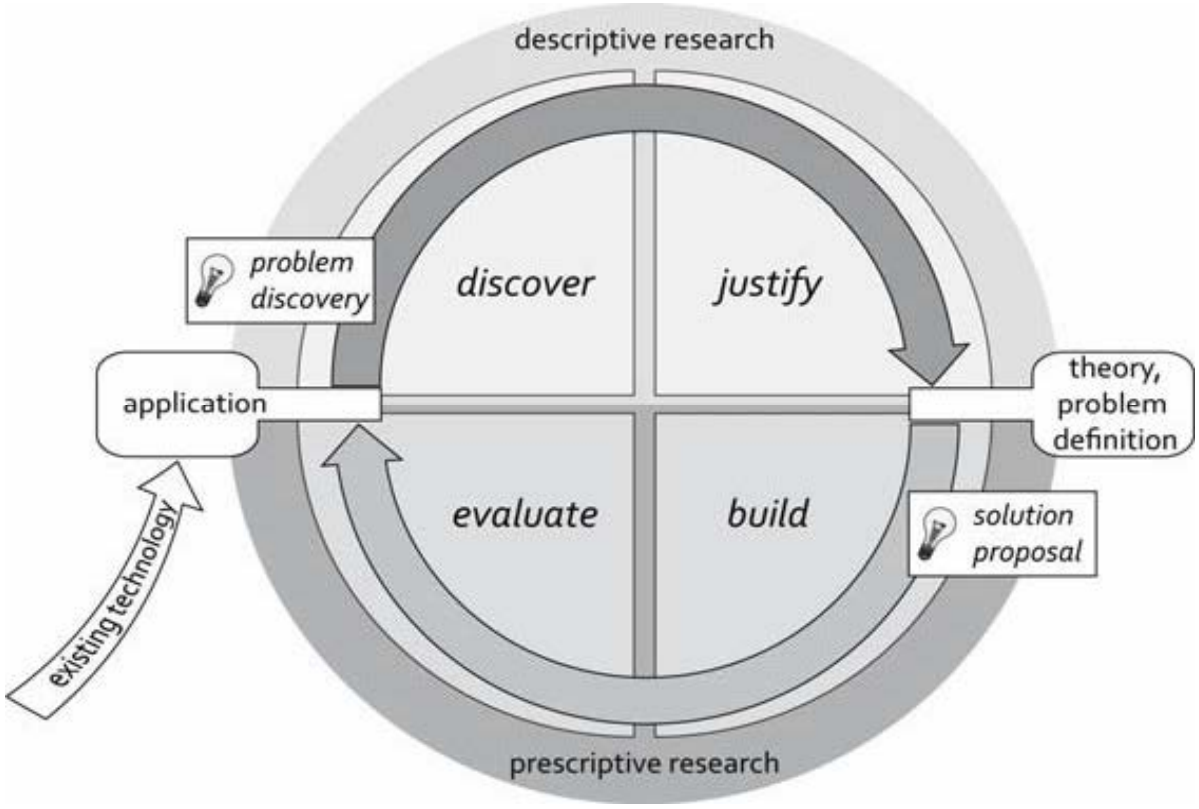


FIGURE 2.4 HEVNER'S RESEARCH CYCLE

because the nature of the research involves finding a solution to a present problem (energy inefficiency in cloud computing) which requires a solid understanding of the root cause of it in order to solve it.

The descriptive part of the research cycles involves two research activities: **discovering** and **justifying**. The discovering activity in this research involved reading from the current literature and collecting information related to what other researchers have found with regards to the actual problem which is 'in our research', i.e. the energy inefficiency in cloud computing. This part of the research provides sufficient evidence of the problem existence and therefore generates the motivation towards solving the research problem.

The second research activity of the descriptive part is the justifying activity. The justifying activity leads to understanding the cause of the problem and to factors that contribute towards it. In this research, the justifying activity also involves obtaining deep understanding and knowledge of the technical details involved in the current system model of cloud computing. The Justifying activity aims at obtaining deeper knowledge of the problem root causes and to understand why such problem exists. Once the justifying activity is completed, the entire image of the problem becomes clear and the information obtained will be used for the prescriptive part of the research cycle. In our research, the justifying activity was conducted based on the current literature and on what other researchers have found and done for achieving sufficient trade-offs between energy consumption and quality of service in the field of cloud computing. Also, an accurate description of the current system model of cloud computing and how it works was one of the major targets that was achieved during the justifying part of the research.

The second part of the research cycle presented in figure 2.4 is the prescriptive part. It involves two corresponding research tasks linked to what has been done in the previous part of the research cycle. In the prescriptive part, two activities are involved: a **building activity** and an **evaluating activity**. The building activity in our research involved designing the new cloud system model and architecture, in which our proposed architectural components were built and employed in the system in response to the issues of the current system model studied during the first part of the research. The process of building the system is fully dependent on what has been obtained during the previous part of the research, as the new (proposed) system model and architecture are intended to overcome the problem which has been discovered and justified during the first part of the research.

The evaluating activity in our research involved studying the behavior of the UPAPS framework in terms of energy consumption. This process is performed through extensive simulation studies. Due to the nature of this research which involves designing a new cloud-based system model, the validation/evaluation of the new system model happens by designing cloud scenarios and running them according to both; the current cloud system model and the newly designed model with the goal of comparing results of both simulation trials and to find out whether the new system model achieves the desired outcomes and research objectives or not. The success of the research outcomes is judged based on the comparison between simulation results for both system models, because the main objective of this research is to develop a new cloud system model that reduces the energy consumption in cloud datacenters without affecting the quality of service delivered, as well

as adding more means of elasticity to the field of cloud computing within the context of the pay-as-you-go model.

The materials used for this research include several heuristic algorithms proposed by other researchers. These Heuristic algorithms will be used to design the cloud scenarios. Each heuristic algorithm will be used to generate a service scheme offered from the cloud service provider to the cloud clients. Heuristic algorithms are discussed later in chapter 3 of this thesis.

Figure 2.5 presents the summary of the methodology followed for this research.

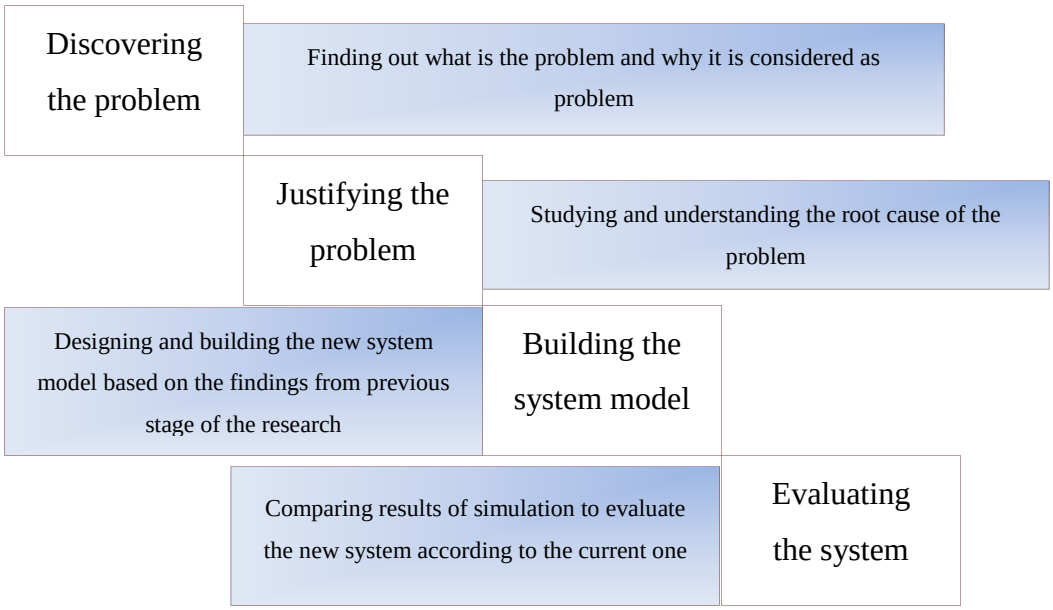


FIGURE 2.5 RESEARCH METHODOLOGY STEPS

2.3 Energy consumption issue

Computing is always in state of constant change as it is witnessed by the breakthrough taking place in the field [28]. Cloud computing is a young phenomenon, and it is suffering through the growing pains typical of its age [29]. Although cloud computing has been widely adopted in various fields in today's industry, the research on cloud computing is still at an early stage as many existing issues have not been yet fully addressed [30].

There are many threats and challenges in cloud computing that are currently being targeted by researchers to overcome them such as security, complexity, reliability and other data related issues.

One of the major issues in cloud computing technology is the energy consumption. The widespread use of cloud computing services is expected to increase the power consumed by ICT equipment in cloud computing environments rapidly [5]. Jayant Baliga and co-authors, from the University of Melbourne in Victoria/Australia, have conducted a research to investigate the use of cloud computing for three different services: storage services, software services, and processing services in public and private systems. Results showed that cloud computing is not always the greenest option in terms of power consumption [31]. The increasing demand on cloud computing infrastructures and services has become a major environmental issue due to the amount of power it requires [5].

Power consumption is today considered one of the most important issues in cloud computing that needs to be addressed. The major aim of this thesis is to propose a solution for the problem of inefficient energy consumption in cloud computing.

One of the major causes of energy inefficiency in cloud computing is the idle power wasted when datacenters and servers run at low utilization [32]. Recently, work conducted by Shin-ichi Kuribayashi on cloud computing has identified a need for collaboration among servers, communication networks, and power networks in order to reduce the total power consumption by the entire ICT equipment in a cloud computing environment [5]. Since the reliability of computing resources is one of the characteristics of cloud computing, the underutilization of computing resources has become a major issue and root cause of the inefficiency of energy consumption in those computing configuration. Servers may run on 10% utilization while they consume energy at their maximum consumption capabilities. The underutilization issue in cloud computing has been a major problem to overcome recently.

Datacenter energy savings can come from both; hardware and software through accurate resource management [32]. In [5], the authors give an example of where accurate management of computing resources is needed for both processing time and network speed where both of them have strong impact on the energy consumption rates, the authors say: 'slowing the processing of a server to reduce its power consumption rate can prolong not only its processing time, but also the bandwidth holding time in the network, this increases the power consumed by the network. Conversely, raising the processing speed of a server increases its power consumption but reduces the processing time, and consequently reduces the power consumed by the network'.

Similarly, the issue of underutilization of computing resources requires an accurate computing resources management framework in order to achieve efficient levels of energy consumption rates without negatively affecting the Quality of Service delivered. The

following section of this chapter focuses on the policy-based network management approach that was proposed by several researchers for the goal of improving the management of computing networks.

2.4 Policy-based network management (PBNM)

Policy-Based Network Management (PBNM) is one of the active research topics that have been driven by the great complexity inherent in the administration and management of today's networking and telecommunications systems [33]. It is today attracting considerable research focus as an empowering technology for managing large scale, heterogeneous information systems and communications infrastructure [34].

PBNM has attracted significant attention both from industry and the academic research community in recent years; it has been recognized that PBNM can effectively provide good means to solve the puzzle of integrated IP/telecom management [35].

PBNM is based on defining a set of global rules, according to which a network or distributed system must operate [36]. In [37], PBNM is defined as the technology that provides the tools for an automated management of networks. It focuses on delivery of services to users and applications rather than devices and interfaces, and thus enables a holistic management of the network.

A research conducted in [38] aimed at providing an overview of how the policy management for autonomic computing (PMAC) platform works and manages networked

systems. The authors intended to demonstrate the concept and the technical issue of the management model on networked systems such as cloud computing systems. The outcomes of the study revealed the ability of policy-based network management in reducing the burden on the human administrator in such networks by providing systematic means to create, modify, distribute, and enforce policies for managed resources [39].

In [40], a research was conducted on the policy-based network management system and its usability for network administration purposes. The main objective of the study was to demonstrate how a network administration can be simplified using a policy-based network management system. It also intended to express the main framework-related issues encountered and those that need to be considered when developing policy-based management systems such as the critical issue of policy conflicts. The policy conflict issue occurs when two or more policies are due to be enforced simultaneously due to satisfactory conditions for both of them at the same time. The outcome of the research indicated that the PBNM framework can be greatly beneficial towards simplifying the management of networks hence achieving more accuracy in network management.

2.4.1 Policy-Based Network Management Architecture

The general policy-based network management framework we present is considered as an adaptation of the Internet Engineering Task Force (IETF) policy framework [41]. IETF is an open international community of network designers, operators, vendors and researchers concerned with the evolution of the internet architecture and the smooth operation of the internet, more information on IETF is available in [42].

The IETF framework for Distributed Management Task Force (DMTF) is shown in figure 2.6.

The DMTF framework as shown in figure 2.6 consists of four main elements: policy management tool, policy repository, policy decision point, and policy enforcement point. Components of the DMTF framework all work together towards managing a particular distributed system by enforcing policies.

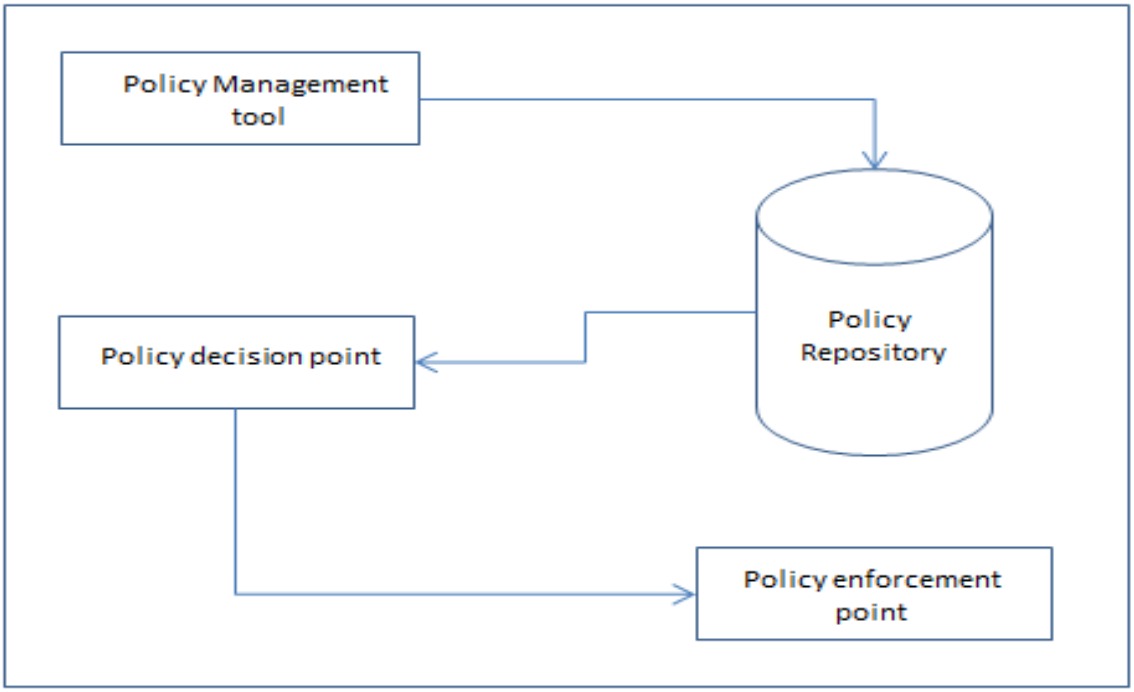


FIGURE 2.6 THE IETF/DMTF POLICY FRAMEWORK

2.4.2 Policy Management Tool

The policy management tool is defined as the component of the framework which allows defining policies to be enforced within the network.

Due to the fact that the policy management tool is not defined by the IETF standards, in this thesis, we focus on this component in terms of how it can leverage the power of policies to simplify the provisioning and configuring of the different devices within the network. The policy management tool simplifies the management functions of a network via two elements of the policy management tool and the policy architecture: centralization and business-level abstraction [40].

Centralization is the process of defining all the device provisioning and configuration at a single point (policy management tool) rather than provisioning and configuring each device itself. This reduces the manual efforts that an administrator puts to configure and provision devices especially in large-scale networks. With a policy management tool, the network administrator inputs the policies needed for network operation into the management tool that populates the repository component which is discussed further in the chapter. For example, in a network that comprises 1000 devices, on average, the administrator needs 10 minutes to configure each of these devices hence he/she needs over a week to complete the configuration work for all devices within the network, while with the policy-based solution, the network administrator requires 15 minutes to populate the repository with appropriate policies, and other components of the framework such as the PDP and the PEP will take care of the rest.

Business-level abstractions simplifies the job of the policy administrator by defining the policies in terms of a language closer to the business needs of an organization rather than in terms of the technical language needed for its deployment.

For example, we assume a case of a network operator that needs to define two levels (high and low) of risk. With the business-level abstraction, it is very simple for an administrator to identify each risk level and to define which level they may map to. The business-level abstractions fully depend on the business needs and the technology that the policies are being defined for, as the business needs of an organization may be satisfied by many different technologies. For example, business needs such as a service level agreement can be satisfied by technologies such as capacity planning [43] or content distribution [44], while a business need such as establishing a secure virtual private network may be satisfied using IP Security (IPSec) [45] or TLS protocol [46].

2.4.3 Policy Repository

The policy repository is the component where policies, which have been created by the policy management tool, are stored. In order to ensure interoperability across products from different vendors, information and policies stored in the repository must correspond to an information model specified by the Policy Framework Working Group [40]. The policy repository may have many different interfaces enabled, in order to allow different types of users to manipulate the contents of the policy database [47].

2.4.4 Policy Decision Point (PDP)

Policy Decision Point (PDP) is the component of the framework that is responsible for interpreting the policies stored in the repository and communicating them to the Policy Enforcement Point (PEP) which is discussed in the following sub-section. PDP works by translating a policy into a form that is understandable to network devices.

PDP is an intermediary point between the point that is responsible for enforcing policies on devices within the network (PEP), and the repository where policies are stored in the system.

2.4.5 Policy Enforcement Point

Policy enforcement point (PEP) is the component in which policies are actually enforced; decisions are actually enforced; policy decisions are primarily made at the PDP [47]. The PEP is responsible for starting the interaction between the components of the entire system, in other words, in case of an event, the PEP formulates a request for the policy decision and sends it to the PDP, and hence the PEP is the component that detects events and requests decisions to be made from the PDP. As soon as a request is formulated and sent to the PDP, the PDP decides which policy to enforce and then forwards it to the PEP to enforce it on the network devices.

2.5 Summary

In this chapter, we first give an introduction and overview of cloud computing technology. Cloud computing services are provided in three different models: SaaS, PaaS and IaaS, each of these service models is provided according to clients requirements. There are five main characteristics of cloud services: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measure services. These characteristics of cloud computing have made such technology increasingly demanded by different types of users such as organizations, individuals and communities. There are four major deployment models of cloud computing: private cloud, community cloud, public cloud, and hybrid

cloud. Each deployment model has different settings according to its objectives and demand. However, despite the great benefits that cloud computing offers, the energy consumption is considered one of the major issues in the field of cloud computing. Inefficiency in power consumption rates leads to environmental issues. The main cause of the energy inefficiency in cloud computing is the issue of underutilization of computing resources.

Secondly, we also introduce the policy-based network management framework based on the current literature. Policy-based network management (PBNM) is considered as a promising management framework that aims at simplifying the management of cloud networks and therefore overcoming the issue of underutilization of computing resources. We present the PBNM framework as an extent to the IETF policy framework. We further explain the functionality of each component of the framework and explain how it works towards managing the network based on policies.

CHAPTER 3: USER PROFILE AWARE POLICY SWITCHING (UPAPS)

In response to the issue of power consumption caused by insufficient usage of computing resources in cloud-based environments, this research presents a new approach that aims at tackling these issues and producing an efficient trade-off between energy consumption and quality of service (QoS) by reducing the energy consumed at the cloud provider side and delivering the demanded QoS. The key concept of the proposed management framework is the consolidation processes towards better utilization efficiency of computing resources. Consolidation processes are done in virtualized cloud environments via migrating virtual machines from a host to another. The consolidation processes involve switching off physical nodes after migrating virtual machines from them to other hosts in a particular datacenter. The processes of virtual machine migration are fully dependent on certain criteria which will be discussed further in this chapter. The UPAPS framework involves a dynamic user profile-aware policy management mechanism that can manage cloud resources in a virtualized-cloud environment. The main goal of the UPAPS framework is to reduce the extra energy consumption caused by an inefficient usage of computing resources without aggressively impacting on the Quality of Service (QoS) provided to users. The proposed approach also aims at improving the cloud infrastructure in terms of business for both cloud provider and cloud users by reducing the operating costs of the cloud infrastructure (datacenters) as well as facilitating more options for users in order to better

manage their ICT costs hence creating more means of elasticity in the field of cloud computing.

In order to successfully achieve the objectives of this research, there are several terms that have to be carefully considered when dealing with a problem related to cloud computing such as Quality of Service (QoS). Quality of Service is a broad topic in distributed systems such as cloud computing and is often referred to as the resource reservation control mechanisms in place to guarantee a certain level of performance and availability of service [48].

This chapter is organized as follows: Section 3.1 presents an introduction and definitions of essential terms related to Quality of Service (QoS) based on the current literature. These terms are to be used for evaluating our proposed approach and model towards overcoming the issue of underutilization of computing resources. Section 3.2 presents and technically describes a group of heuristic algorithms which have been developed and proposed by other researchers in hope to overcome the issue of energy consumption in cloud computing. Heuristic algorithms were proposed by other researchers in the field for the goal of performing virtual machine migrations and therefore physical machines consolidation. Section 3.3 presents our proposed system model for cloud computing networks. It provides technical details on how such model works and how it contributes towards overcoming the issue of energy consumption in cloud computing. It also points out issues with the current system model, and explains how the proposed system model can be a potential solution to overcome these issues. Section 3.4 presents our proposed cloud architecture which works along with the proposed system model towards overcoming the issue of energy inefficiency in cloud computing.

It explains all components involved with in the proposed architecture, and provides details on how each component works towards the objective of the proposed architecture.

3.1 Service Level Agreements (SLA) Metrics

Quality of Service (QoS) is a very important matter in the field of cloud computing and thus meeting its requirements and maintaining it is an essential requirement for the management of any cloud-based environment. In [49], the QoS of a network is defined in a variety of ways and include a diverse set of service requirements such as performance, availability, reliability, security, etc. All these service requirements are important aspects of a comprehensive network service offering. QoS refers to the capability of a network in providing better service to selected network traffic over various technologies such as Ethernet and 802 networks, IP-routed networks which may use any or all of these underlying technologies [50].

QoS is normally delivered in the context of an agreed Service Level Agreements (SLAs) which can be specified in terms of many characteristics such as minimum throughput, availability or maximum response time delivered by the deployed system [8].

In [51], the author states: A successful QoS deployment includes three key phases: strategically defining the business objectives to be achieved via QoS, analyzing the service-level requirements of the traffic classes, and designing and testing QoS policies.

In [52], the network infrastructure must be designed to be highly available in order to successfully implement QoS, and the target for high availability of service is 99.999% up time, with only five minutes of downtime permitted per year. The transmission quality of the network is determined by three factors: loss, delay and delay variation (Jitter).

Loss is a relative measure of the number of packets that were not received compared to the total number of packets transmitted. In order to obtain the highest availability of a network, loss during periods of non-congestion would be essentially zero. Delay is defined as the predetermined amount of time that a packet requires to reach the endpoint after being transmitted from the sending endpoint. Delay variation (Jitter) is the difference in the end-to-end delay between packets.

However, QoS requirements in cloud computing are normally delivered in a form of SLA. SLA characteristics can then vary according to various applications domains and for cloud computing, it is necessary to define independent workload metrics to be used for evaluating the performance of the virtual machines that are running in an Infrastructure as a Service (IaaS) configuration.

In [8], their research was on virtualization in cloud computing, it involved a competitive analysis and proven competitive ratios of optimal online deterministic algorithms for single virtual machine migration and dynamic virtual machine consolidation problems. The authors defined two workload independent metrics that can be used to evaluate the SLA delivered to any VM on IaaS: percentage of time during which active hosts have experienced CPU utilization of 100% (SLATAH), and degradation of performance caused

by VM migration (PDM). For this research, due to the similarity in research objectives and goals, we intend to use the two major SLA metrics proposed by [8] (SLATAH and PDM).

3.1.1 SLA Violation Time per Active Host (SLATAH)

The first metric we choose for our research is the percentage of time in which active hosts have experienced a CPU utilization of 100%. This SLA metric is considered suitable to use in this research because it helps towards validating our proposed approach that involves virtual machine migrations and consolidation processes. The consolidation approach involves migrating virtual machines from a host to another, for the goal of switching off the "home" physical host to a power saving mode to save energy. This requires selecting virtual machines to migrate, and further choosing a "new physical" host to accommodate them. Moreover, virtual machine migrations serve towards balancing the load on physical servers (hosts). For that, the SLATAH metric has been chosen for our research because it observes and indicates to the host that it is experiencing 100% CPU utilization. When a host CPU capacity is being 100% utilized, virtual machines on this particular host might not be fully provided with the required performance level, this leads to performance degradation of virtual machines and hence SLA violations.

The SLA violation Time per Active Host (SLATAH) is calculated mathematically as shown in equation (1):

$$SLATAH = \frac{1}{N} + \sum_{i=1}^{N} \frac{Ts_i}{Ta_i} \qquad \textbf{(1)}$$

where N indicates the number of hosts; Ts_i is the total time during which the host i has experienced the utilization of 100% leading to an SLA violation Ta_i indicates the total time during which host i is in the active state (serving virtual machines).

The SLATAH metric indicates to the violation degree/level to the SLA in terms of controlling and monitoring the status of a particular host. If SLATAH is high, this means the host often reaches the maximum capacity utilization which leads to violation in SLA.

3.1.2 Performance Degradation due to Migration

The second metric is the degradation of performance caused by VM migration (PDM). The reason of choosing the PDM as an SLA metric for this research is due to the nature of the study, as virtual machine migration processes might have a negative impact on the performance of the virtual machines. When a virtual machine is going through a migration process from a host to another, there is a period of time during the process in which the virtual machine goes down. The downtime of the virtual machine has a negative effect on the QoS delivered and therefore on its SLA terms. The PDM metric is calculated mathematically as shown in equation (2).

$$PDM = \frac{1}{M} + \sum_{j=1}^{M} \frac{Cd_j}{Cr_j} \qquad \textbf{(2)}$$

where M indicates the number of virtual machines, Cd_j is the estimate of the performance degradation of the virtual machine j caused by migration, Cr_j is the total CPU capacity requested by the virtual machine j during its lifetime which is estimated as 10% of the CPU utilization in MIPS (Million Instructions per Second) during all migrations of the VMj.

Since both the SLATAH and PDM metrics have similar importance in terms of the level of SLA violation by the infrastructure, they can be combined into one SLA Violation (SLAV) metric which is calculated as shown in equation (3).

$$SLAV = SLATAH \,.\, PDM \qquad \textbf{(3)}$$

3.2 Heuristic Algorithms for Dynamic VM Consolidation

The term “Heuristic” was initially created by the Greeks; its original meaning is “Discover”. Algorithms that either give nearly the right answer or provide solution not for all instances of the problems are called heuristic algorithms. Heuristic algorithms’ validation and complexity is one of the most important topics in the field of computer science today [53].

Heuristic algorithms are today used in computer science to quickly solve complex problems and perform tasks. In order to understand the concept of heuristic algorithms, the Travelling Salesman problem [54] is one of the classic enigmas in computer science for which heuristic algorithms have found many heuristic solutions. More information about heuristic algorithms can be found in [53].

Since the beginning of Genetic Algorithms (GA) and heuristic algorithms research in general, it has become known that parameters and operators have significant impact on the optimization process and the efficiency of such algorithms [55]. Due to the nature of this study, in order to obtain the best results of heuristic algorithms, it is very essential to define parameters and operators which are involved in our research target problem.

For our research, the nature of the problem is fully related to virtual machine migration and consolidation processes hence, choosing suitable heuristic algorithms and defining suitable parameters and operators is highly important to solve this problem.

Policy	ESV ($\times 10^{-3}$)	Energy kWh	SLAV ($\times 10^{-5}$)	SLATAH	PDM	VM migr. ($\times 10^{3}$)
NPA	0	2419.2	0	0%	0%	0
DVFS	0	613.6	0	0%	0%	0
THR-MMT-1.0	20.12	75.36	25.78	24.97%	0.10%	13.64
THR-MMT-0.8	4.19	89.92	4.57	4.61%	0.10%	17.18
IQR-MMT-1.5	4.00	90.13	4.51	4.64%	0.10%	16.93
MAD-MMT-2.5	3.94	87.67	4.48	4.65%	0.10%	16.72
LRR-MMT-1.2	2.43	87.93	2.77	3.98%	0.07%	12.82
LR-MMT-1.2	1.98	88.17	2.33	3.63%	0.06%	11.85

TABLE 3.1 SIMULATION RESULTS OF THE BEST ALGORITHM COMBINATIONS AND BENCHMARK ALGORITHMS

In [8], the authors have proposed unique adaptive heuristics that are based on analysis of historical data on the resource usage for performance and energy efficient dynamic consolidation of VMs. The authors have proposed five host overloading detection algorithms: Static Threshold VM allocation policy (THR), Inter Quartile Range (IQR), Median Absolute Deviation (MAD), Local Regression (LR), Local Regression Robust (LRR), and three VM selection algorithms; Minimum Migration Time (MMT), Random Selection (RS), and Maximum Correlation (MC).

Table 3.1 illustrates the characteristics of these different algorithm combinations (median values).

All combinations of host overloading detection algorithms and selection algorithms have been extensively simulated on large-scale experimental setups with the goal of comparing their efficiencies in terms of the trade-offs between power consumption and quality of service violation (SLAV).

Based on the results illustrated in Table 3.1, it is clear the difference in the efficiency among all the proposed consolidation algorithms in terms of SLA violation and power consumption ratios. The dynamic VM consolidation algorithms (NPA and DVFS) significantly perform better than static allocation policies; however, they lead to greater power consumption rates in comparison to all other consolidation algorithms.

There is also a proportional relationship between the number of virtual machine migrations and the SLATAH metric for all the static allocation policies. The reason behind the variation of efficiency levels lies on the details of each algorithm and the metrics that each algorithm uses to select and allocate VMs from one host to another. According to the results in Table 3.1, LR-MMT-1.2 produces the best trade-off between power consumption and SLA violations due to the relatively small number of VM migrations, while the efficiency in terms of the same trade-off decreases by going up in the Table as LRR-MMT-1.2 is the second best algorithm followed by MAD-MMT-2.5 and so on.

In this research, we intend to employ the top three static allocation policies which are LR, LRR and MAD, one dynamic VM consolidation algorithm (DVFS), and MMT as the virtual machine selection policy proposed by [8]. We aim at using these policies for the goal of designing our proposed system model and validate it towards overcoming the issue

of underutilized computing resources in cloud-based environments and to eventually reduce the amount of energy consumed without violating the SLA.

3.2.1 Static Allocation Policies

3.2.1.1 Local Regression (LR)

The main idea behind the Local Regression algorithm is fitting simple models to localized subsets of data for the goal of building up a curve that approximates the original data. Local Regression is based on the Loess method proposed in [56].

The observations (x_i, y_i) are assigned neighborhood weights using the *tricube weight* function presented in (4)

$$T(u)= \begin{cases} (1-|u|^3)^3 & \text{if } |u| < 0 \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

Assuming that $\Delta i = (x) = |x_i - x|$ is the distance between x and x_i and $\Delta i\ (x)$ denotes these distances ordered from shortest to longest, then neighbourhood weight for the observation $(x_i\ ,\ y_i)$ is calculated using equation (5) below.

$$w_i\ (x) = T\ \left(\frac{\Delta i\ (x)}{\Delta_{(q)}\ (x)}\right) \qquad (5)$$

For x_i such that $\Delta i\ (x) < \Delta q\ (x)$ where q denotes to the number of observations in the subset of data localized around x, the size of the subnet is then defined by a parameter of the method called bandwidth. For example, if the degree of the polynomial fitted by this

method is 1, then the parametric family of functions is $(y) = (a)+(bx)$. The line is fitted in the observations subset using the weighted least squares method with weight $w_i\,(x)$ at (x_i, y_i). The values of (a) and (b) can be calculated by the equation (6).

$$\sum_{i=1}^{n} w_i\,(x)(y_i - a - \mathrm{b}x_\mathrm{i})^2 \qquad (6)$$

This approach is used to fit the trend polynomial and find the last k observations of the CPU utilization where $(k) = \lceil {}^{q}/_{2} \rceil$. The polynomial is fit for one single point and the last observation of the CPU utilization is the right boundary (x_k) of the dataset. In [56], all fitted polynomials of the degree 1 typically distort peaks in the interior of the configuration of observations, whereas polynomials of degree 2 remove the distortion but result in higher biases at boundaries. This is the reason behind choosing polynomials of the degree 1 as it reduces the bias at the boundary.

Assuming that (x_k) is the last observation, and (x_1) is the (k^{th}) observation from the right boundary. For that, (x_i) is assumed to satisfy $(x_1) \leq (x_k) \leq (x_k)$, then $(x_k) = \Delta i\,(x_k) - (x_i)$, and $0 \leq \frac{\Delta i\,(x_k)}{\Delta 1\,(x_k)} \leq 1$. Hence, the function of the tricube weight can be simplified as $T^* u = (1 - u^3)^3$ for $0 \leq u \leq 1$, and the weight function can be formulated as shown in equation (7)

$$w_i\,(x) = T^*\left(\frac{\Delta i\,(x_k)}{\Delta 1\,(x_k)}\right) = \left(1 - \left(\frac{x_k - x_i}{x_k - x_1}\right)^3\right)^3 \qquad (7)$$

In Local Regression (LR), with consideration to the explained method which is derived from Loess, we find new trend line $\hat{g}(x) = \hat{a} + \hat{b}_x$ where each new observation is used to estimate the next observation $\widehat{b}(x_k + 1)$. The algorithm therefore decides whether the host is overloaded and there is a need for virtual machine migration or not with consideration to the following inequalities shown in (8)

$$s.\widehat{g}(x_k + 1) \geq 1, \qquad x_k + 1 - x_k \geq t_m \qquad (8)$$

Where $s \in R^+$ is the safety parameter; and t_m is denotes to the maximum time required for a migration of any of the virtual machines allocated to the host.

3.2.1.2 Local Regression Robust (LRR)

The Robust local regression (RLR) is an extension to the Local Regression described in the previous subsection. Since the (LR) described in the previous section (3.1.3.1.a) is considered vulnerable to outliners and extreme readings due to leptokurtic or heavy-tailed distribution, the authors in [56] have proposed the addition of the robust estimation method *bisquare* to find the "*least squares*" for more accurate parametric fitting in the entire distribution of observations. The starting fitting process is done with weights defined by the tricube weight function mentioned earlier. The fit then gets evaluated at the x_i to get the fitted values $\hat{y}_i$, and the residuals $\widehat{\epsilon_i} = y_i - \hat{y}_i$. In the next step, every observation (x_i, y_i) is assigned an additional robustness weight r_i, whose value depends on the magnitude of $\widehat{\epsilon_i}$. Every observation is assigned the weight $r_i w_i(x)$, where r_i is defined as shown in equation (9)

$$r_i = B\left(\frac{\widehat{\epsilon_i}}{6_s}\right) \qquad (9)$$

where ‘s’ denotes to the median absolute deviation for the least-square fit or subsequent weighted fit as shown in equation (11), and B(u) denotes to the bisquare weight function shown in (10).

$$\text{B(u)} = \begin{cases} (1-u^2)^2 & \text{if } |u| < 1 \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

$$s = median|\widehat{\epsilon_i}| \qquad (11)$$

The next observation is then estimated using the estimated trend line, by applying the method described in section (3.1.3.1.a) for the goal of deciding whether the host is overloaded or not with consideration to the inequalities shown in equations (8).

3.2.1.3 Median Absolute Deviation (MAD)

One way of deciding whether there is a need for virtual machine migration can be based on utilization threshold of the physical host. This can be done by simply setting upper and lower utilization thresholds for hosts and keeping the total utilization of the CPU by all the virtual machines between these thresholds. If the utilization threshold of a host falls below the lower threshold, all virtual machines has to be migrated from this host in order to switch it off to save energy. If the utilization threshold exceeds the upper threshold, virtual machines have to be migrated to another host to avoid potential violations to the SLA.

However, due to the fact that fixed threshold values are not suitable for environment with dynamic and unpredictable workloads, there is a need for other techniques that can

automatically adjust the utilization threshold of hosts based on statistical analysis of historical data collected during the life time of virtual machines. One solution toward overcoming the issue of setting up threshold values for hosts is relying on the strength of deviation of the CPU utilization. When deviation is high, the value of the upper utilization threshold is low because CPU utilization is vulnerable to reach the maximum utilization value and therefore causes a SLA violation.

The Median Absolute Deviation (MAD) can be defined as a measure of statistical dispersion. Its estimates scale more robustly than the sample variance or standard deviations as it interacts more accurately with distribution without a mean or variance, such as Cauchy distribution [57]. The MAD is a robust statistic; it is more resilient to outliers in datasets than the standard deviation. In standard deviation, distances from the mean are squared hence, large deviations are weighted more heavily, and thus it can be greatly influenced by outliers. In the MADs measure, the distances magnitude of small number of outliers is not relevant.

For instance, we assume a dataset of $(X_1, X_2, X_3, X_4 \ldots . X_n)$ the MAD is defined as the median of the absolute deviations from the dataset's median as shown in the following equation

$$MAD = median_i \left(\left| X_i - median_j(X_j) \right| \right) \qquad (12)$$

In other words, the MAD is defined as the median of the absolute values of a dataset. Hence, the upper utilization threshold can be calculated using the following equation

$$T_u = 1 - s \,.\, MAD \qquad (13)$$

where $s \in R^+$ the parameter of the method that defines how aggressively the system consolidates virtual machines. Moreover, the parameter *s* facilitates the adjustment of the safety of the method; the lower 's' decreases the energy consumption but causes higher SLA violation due to aggressive consolidation of virtual machines.

3.2.2 Dynamic VM consolidation algorithm

3.2.2.1 Dynamic Voltage Frequency Scaling (DVFS)

The dynamic voltage frequency scaling (DVFS) is a commonly used technique to save power on a wide range of computing systems, from embedded, laptop and desktop systems to high-performance server-class systems [58]. DVFS is a very popular power management technique where the clock frequency of a processor is decreased to allow a corresponding reduction in the supply voltage.

DVFS offers great opportunities to dramatically reduce energy consumption by adjusting both voltage and frequency levels of a system according to the changing characteristics of its workloads [59].

DVFS is able to reduce the power consumption of complementary metal-oxide-semiconductor (CMOS) integrated circuits [60], such as a modern computer processor, by reducing the frequency at which it operates as shown in the following equation [58]

$$\mathrm{P} = CfV^2 + P_{static} \qquad (14)$$

where C is the capacitance of the transistor gates which depends on feature size, *f* denotes to the operating frequency, and *V* denotes the supply voltage.

The main idea of the DVFS technique is to intentionally scale down the CPU performance, when it is not fully utilized, by decreasing the voltage and frequency of the CPU. The DVFS Algorithm can be applied in various applications, however, due to the nature of this study, and its research objectives which are fully related to virtual machine migration, we here present the four major processes that DVFS performs in order to reduce the energy consumption under the umbrella of virtual machine migration in virtualized environments.

The DVFS algorithm adjusts the hosts' energy consumption according to their CPU utilization. There are four main processes that DVFS performs in order to reduce the total energy consumed in a datacenter:

I. The first step involves getting signal acquisition of the system load (CPU utilization)

II. The system load is then used to predict the amount of energy it requires in the next period of time

III. The predicted amount of energy is then transformed into the desired frequency, and therefore the clock from the chip set is changed accordingly.

IV. Finally, the new frequency is used to calculate the new voltage, and then the power management module gets notified to adjust the voltage of the CPU.

More information on DVFS algorithm can be found in [61].

3.2.3 Virtual machine selection policy

3.2.3.1 Minimum Migration Time (MMT)

The minimum migration time (MMT) policy migrates virtual machines according to the length of time they require to complete their migration processes. MMT works by listing all virtual machines and sorting them according to the time they require to complete their

migration. The virtual machine that requires the minimum time to complete its migration is selected for migration.

The first step a MMT algorithm takes in its operation is listing virtual machines that are running in one host and are eligible for migration. Once all virtual machines are listed for migration, the algorithm looks for virtual machines which require the least time to complete their migration process and selects them for migration to another host.

The process of finding the virtual machine migration time is fully related to the RAM utilized by the virtual machine. The migration time for a virtual machine is estimated as the amount of RAM utilized by a virtual machine divided by the spare network bandwidth available for the home host.

Let's assume that host j is currently allocated for a number of virtual machines running on it, and V_j is the set of virtual machines running on the host j. The MMT algorithm selects the virtual machine that satisfies the conditions formalized in the following equation

$$\mu \in V_j \left| \forall_a \in V_j, \frac{RAM_\mu(\mu)}{NET_j} \leq \frac{RAM_\mu(a)}{NET_j} \right. \qquad (15)$$

where $RAM_\mu(\mu)$ is the amount of RAM that is currently utilized by the virtual machine a, and NET_j is the spare network bandwidth available for the host j.

3.3 Issues with the current system model

One of the main advantages that cloud computing offers to users is the pay-as-you-go elasticity [6]. Cloud computing allows users to pay for what they actually use based on

time, storage capacity, and other factors that can be agreed on between the cloud provider and users instead of paying for what they can potentially use. However, the current implementation of virtualized cloud computing environments is not yet efficient due to the issue of underutilization of computing resources. Underutilization of computing resources causes extra energy consumption and therefore increases the operating cost of computing resources in the cloud. Nevertheless, the underutilization of computing resources can also be a disadvantage for cloud users; for example, users may demand the highest available QoS from the cloud provider for a certain time window during the day, while they do not need the same QoS at other time window. According to the current system model, for users, in order to satisfy their requirements during the time window in which the highest available QoS is needed, they need to purchase highest available quality of service throughout the day hence they pay for what they do not really demand for their individual's/organization's needs in one full day.

3.4 The UPAPS Framework

The main contribution of this thesis is to propose a new framework for managing a group of green virtual machines migration policies in a virtualized cloud environment, with the goal of overcoming the issue of underutilization of computing resources which causes inefficiency in power consumption while still satisfying user requirements in cloud datacenters. In order to design an accurate management framework for any cloud-based environment, we propose a new system model that uses several adaptive heuristic algorithms for dynamic virtual machines consolidation as proposed in [8] and that can be managed using a policy-based management framework.

The objective of our proposed system model is twofold; firstly, it addresses the issue of underutilized computing resources by dynamically consolidating VMs, hence reducing the power consumption. Secondly, it involves users in the management of their own profiles and facilitates more flexibility in terms of service schemes that a cloud provider can offer to users; this helps users decide what and how a cloud service can best suit and satisfy their organizations' or individuals' needs with consideration to costs and time benefiting both cloud users and cloud providers.

The main feature of our UPAPS framework is that it allows users to choose the quality of service level that they need for a particular time-window in their user profile. The system allows users to switch from a particular quality of service mode to another; for example, a user might demand the highest available QoS during business hours and the lowest available QoS with cheaper costs after business hours with the goal of minimizing the costs of their cloud usage. Enabling users to manage their required QoS can vastly contribute towards addressing the issue of resources underutilization, because it decreases the operating costs of the cloud by eliminating the energy consumed due to underutilized resources.

In order to illustrate our proposed system model, we introduce an example of a cloud network which provides four service schemes to users at four different price levels. Each service scheme has a different QoS level; hence each service scheme has a different price for users. The differentiation between service schemes happens according to the SLA metrics defined and discussed in section (3.1).

The price of the service scheme is inversely correlated with the SLA violation. In other words, the service scheme that provides QoS with a higher SLAV has relatively cheaper price than other service schemes providing service with a lower SLAV metric. The reason behind the low-cost of any service scheme that provides QoS with SLAV is due to the low cost of producing it on the cloud infrastructure end, because the higher SLAV indicates higher VMs consolidation rates and therefore less power consumption. Unlike previous models in which cloud providers pay high power bills just for delivering the highest available QoS for users who do not necessary need it.

Each service scheme is generated using an adaptive heuristic consolidation algorithm proposed in [8]. We opt to design our proposed system with three service schemes that provide QoS with SLAV on different levels, and another scheme which provides the highest possible QoS without any degradation of service.

The process of deciding the SLA for each service scheme happens through simulating the heuristic algorithm used for that scheme to find out the percentage of performance degradation caused by the algorithm due to aggressive VMs consolidation processes. Based on the Quality of service degradation percentage, SLA terms are set and the price for that particular service scheme is decided accordingly. For example, a particular service scheme denoted as Service C provides service with 8% performance degradation, while another service scheme denoted as Service B provides service with 4% performance degradation. The purchasing price for Service C will be relatively cheaper than the price for Service B. This allows users to minimize the cost of their IT services during the time when quality of service is not a big concern for them.

For example, a customer might not demand high quality of service during the night time; they might only demand email service, hence, a delay of service or any sort of performance degradation might not have an impact on their business. Therefore, a cheaper price for their cloud service is beneficial to them in terms of business as it reduces their business operating costs. On the other hand, the cloud provider obtains the benefit of reducing the amount of power consumed by datacenters. The consolidation processes that take place due to the heuristic algorithm (Policy) leads to a reduction of power consumption as seen in Table (1); this eventually generates significant savings in terms of power consumption and operating costs.

3.4.1 Policy-Based Management Approach

Once all service schemes are decided (heuristic algorithms), it becomes compulsory to employ a management framework that enables cloud users to switch from a service scheme to another without the necessity of having a manual administration at the cloud provider end. For that, we propose the adoption of a Policy-Based Network Management framework (PBNM) that can automatically switch policies according to the SLA signed between cloud service provider and cloud users.

Policy-based network management (PBNM) is a promising solution for managing heterogeneous networks. It addresses the requirements for providing flexible and dynamic management and deals with the escalating size and complexity of modern systems [62]. Policy-based network management aims at simplifying the complex management tasks of large scale systems, since the system based on a number of policies monitors the network and automatically enforces appropriate actions [40]. As discussed in section (2.4), the

PBNM framework consists of four main components namely: policy management tools, policy repository, policy decision points and finally policy enforcement points. Policy management tools are used by the administrator to define policies to be enforced within the network [40].

3.4.2 User-Profile-Based Differentiated Services Architecture

In order to implement our proposed system model, we propose a new architectural component called the User Service Profile (USP) which is a database that contains instructions that a user chooses for the management of his/her profile in order to satisfy their needs. User Service Profiles (USPs) refer to the sequence of policies that are to be

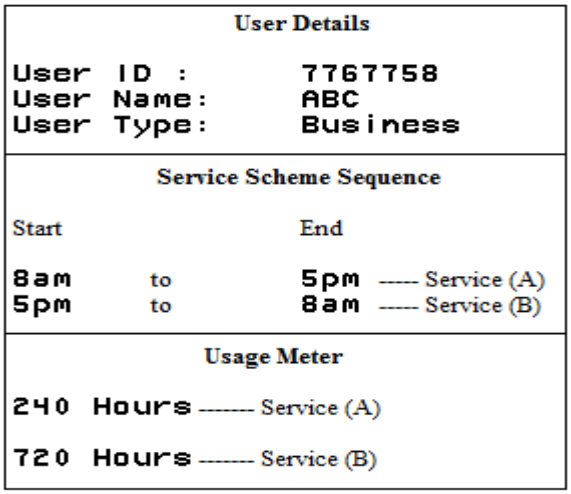


FIGURE 3.1 EXAMPLE OF USER SERVICE PROFILE (USP)

used according to time, work load, or other metrics decided by users within the available options offered by the cloud provider. For example, a user can decide to purchase Service A during business hours (from 8am to 5pm), switching to Service B for the rest of the day; these requirements are coded into the system by users via a user interface supplied by the

cloud service provider. A user service profile decides the SLA terms agreed on between the cloud provider and the user.

The policy repository component stores all policies produced by the policy management tool in the system. All USPs are also stored in the policy repository. The cloud system deals with users according to their profiles. Each rule in a USP has a policy code in the repository to comply with; for example, at 5pm the system automatically invokes a policy from the repository to be enforced by the policy enforcement point to switch the user from Service A to Service B as per our previous example, and at 8am another policy switches the user from Service B to Service A and so on. Figure 3.1 illustrates how a User Service Profile can be generated.

The Policy Enforcement Point (PEP) is the component that is responsible for enforcing

User Profile Aware Policy Switching (UPAPS)

```
1   Input: Usp List  Output: service switching
2   UspList.ListAllUsps()
3   foreach Usp in UspList do
4      Start time      start
5      End time        end
6       foreach Usp in UspList do
7          if timeNow   (start,end) then
8             current service remains
9            if timeNow   (start,end) then
10              switch service to next time
11              window service
12    return service switching
```

FIGURE 3.2 USER PROFILE AWARE POLICY SWITCHING (UPAPS)

policies throughout the network (cloud system). The PEP uses the Policy Decision Point (PDP) as an intermediary in order to communicate with the repository. The PDP is responsible for interpreting the policies stored in the repository and communicating them to the PEP.

To solve the problem of policy switching, we propose the User Profile-Aware Policy Switching (UPAPS) algorithm presented in Figure 3.2. The metric used in our proposed algorithm is time; however, the same algorithm can be used with other metrics such as work load, utilization or just by modifying metrics in the algorithm and stating the preferred one. According to the pseudo-code for our proposed algorithm presented in figure 3.2, the scheduler invokes the method that lists all User Service Profiles (USPs) which are running on a host.

In steps 2 to 4, the scheduler sets the parameters as metrics to be used. The key parameters according to our proposed algorithm (UPAPS) are the start time and the end time. The start time indicates the start time of a particular time window for a particular service scheme according to the USP of each cloud user, while the end time indicates the end time of the same particular time window.

Using the same example mentioned earlier, a user might decide to purchase Service A from 8am to 5pm, the start time for Service A for this particular user is 8am, while the end time for the same service for the same user is 5pm; therefore this particular user during this particular time window (8am to 5pm) should be provided with Service A. Using the parameters defined in steps 2 to 4, and in steps 5 to 7, for each user service profile (USP), the scheduler checks if the current real-time falls between the start time and end time for

the current service (time window) provided for a particular user at that particular moment. If the current time falls in that time window, the scheduler lets the current service continue without switching it. In step 8 to 11, if the scheduler finds that the current real-time does not fall within the current time window for the service being provided to a particular user, it looks for the second time window in which the current real-time falls within, and then switches the current provided service to the service for the proper time window such as Service B.

3.5 Proposed System Architecture

In order to illustrate our proposed system architecture, we find it useful to draw an example scenario of a cloud service provider with two major users: a university and a bank. As seen in figure 3.3, the cloud provider has two major architectural units which are part of the policy-based network management framework namely; Policy management tool and repository, both are connected to the main policy decision point (PDP) which belongs to the cloud provider's system architecture, it is also called the mother PDP. The main role of the mother PDP is deciding policies to be invoked from the policy repository unit to be distributed and enforced in further steps in the system. Each user, in order to manage their own network, has an internal PDP as a gateway between them and the cloud service provider. The users' main PDP is connected to the mother PDP in order to communicate with the cloud provider system and receive policies to be further distributed to the internal PDPs within their own network. In our example scenario, the bank receives policies from the mother PDP through its local main PDP which is as mentioned earlier considered as the gateway to the cloud service provider. The main local PDP in the bank has another policy

repository that is created internally by the bank in order to store policies on the local system/network; however, it was not drawn on the figure with the goal of simplifying the example and to better illustrate the main concept of our proposed system architecture. The main local PDP forwards policies to another PDP within the local system that is responsible for distributing these policies to users according to their USPs. Each user has a Policy Enforcement Point (PEP) assigned to it, in other words, each user receives services and works according to policies delivered to it through its assigned PEP. In our example, a user can be a local network or a group of computers connected within one network that can be a particular branch in the bank or a department …etc. Policies enforced on each user are decided by the PDP according to the user's USP.

As seen in Figure 3.3, each user has a PEP and USP in order to be part of the system. Each device runs an instance of PEP and its corresponding USP.

The PEP reads instructions from the USP and communicates with the PDP accordingly; the PDP retrieves policies from the policy repository and sends them to the PDP. The PDP responds to the PEP and supplies the PEP with the policies to be enforced on its particular user.

The second user in our example in figure 3.3 is a university. The reason for having two PDPs after the main local PDP is due to the fact that a university (as a big enterprise) usually has more than one operating environment or platforms running within its networks, moreover, it is common for such enterprise to have more than one location (computers/offices), hence, we aimed to draw out example with two PDPs to show the

flexibility that our system model provides in terms of satisfying requirements that any network system might demand.

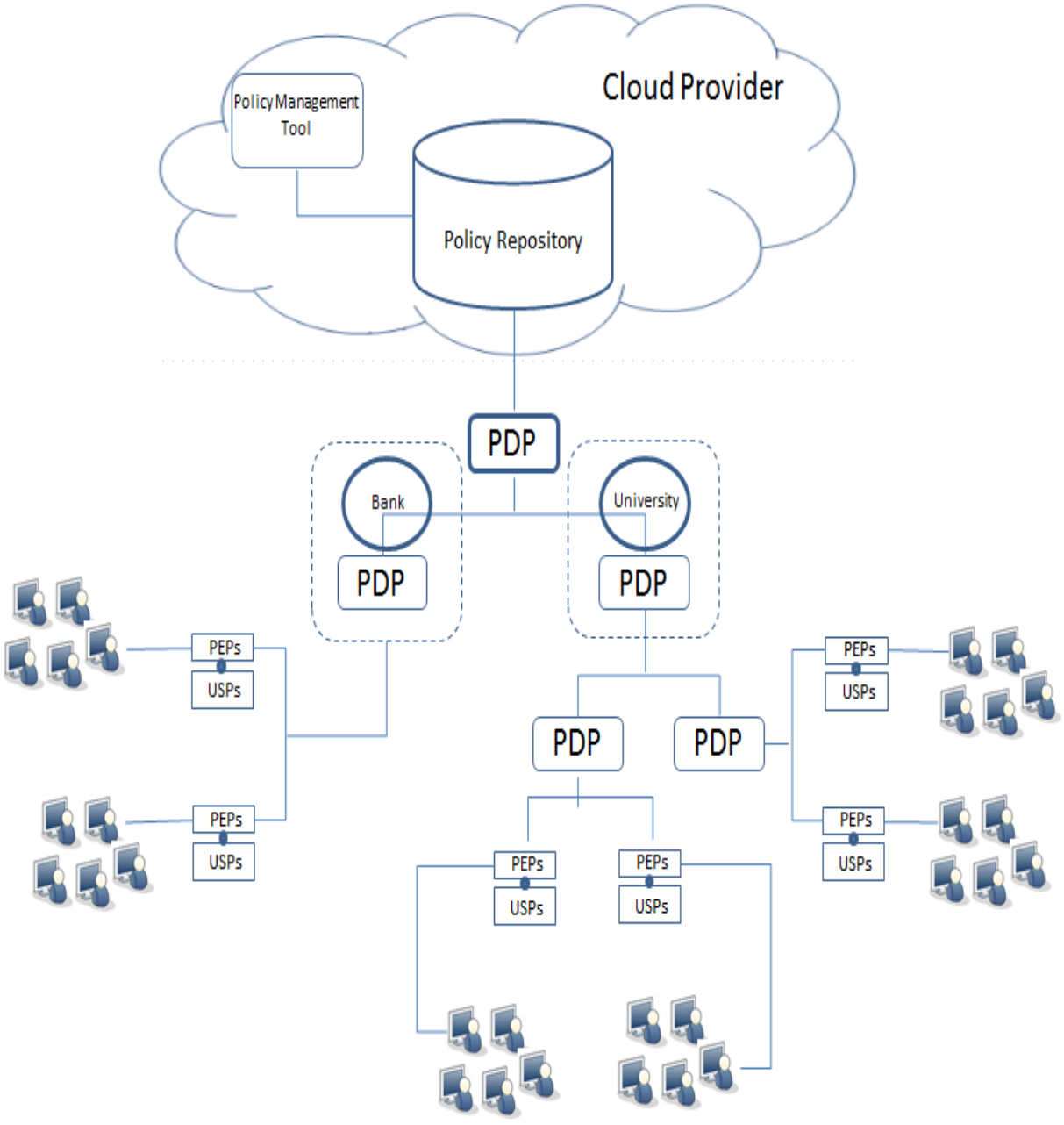


FIGURE 3.3 PROPOSED SYSTEM ARCHITECTURE

3.6 Summary

In this chapter, we first give a detailed introduction on the main goals of the proposed approach and system model. The proposed approach aims at tackling the issue of underutilization of computing resources in cloud-based environments using virtual machine migration and therefore host consolidation. The main idea of the proposed approach is to employ a group of heuristic algorithms in which each of them represents a service scheme that a cloud provides to users using the policy-based network management framework. Each heuristic algorithm causes a certain level of SLA violation. Two SLA violation metrics proposed by others have been identified to be used for validating our proposed approach. We further provide an overview on the issues with the current cloud system model, which our proposed system model aims at overcoming. Furthermore, we also draw our proposed system architecture in a context of a case scenario. Two architectural components have been proposed and employed in the proposed cloud architecture: User Service Profile (USP) and User Profile Aware Policy Switching algorithm (UPAPS).

CHAPTER 4: SIMULATION STUDIES

In this chapter, we present extensive simulation studies which we have conducted on our proposed cloud computing system model and architecture. We have chosen the CloudSim toolkit [26] as a simulation platform for this research. The proposed cloud system model is studied and validated through different network scenarios, for the goal of testing its efficiency in terms of energy consumption reduction, as well as its ability towards adding more elasticity to the technology of cloud computing in the context of provider-user business relations. For the validation of our proposed system model, two SLA metrics were defined to be used: SLATAH and PDM which were proposed in [8], and the simulation processes involved investigating and employing several heuristic algorithms proposed by other researchers for the goal of validating the proposed system model. The validation of our proposed system model is done based on defined SLA metrics in different network scenarios. The proposed system model has proved ability in handling several heuristic algorithms in the context of a policy-based management framework. The proposed system model has shown contributions in the field of cloud computing: firstly, it contributes towards decreasing the energy consumption in datacentres; secondly, it gives clients the facility of managing their own service schemes according to the needed service.

The simulation studies were conducted in several stages; the first stage involved simulating cloud scenarios according to the current system model, and next simulating the same cloud scenario according to the proposed system model. The proposed system model has proved to provide improvements in energy efficiency, and showed more elasticity that cloud clients

can enjoy in terms of budget control on their ICT services as well as convenience in changing the scheduling of their service schemes according to their needs and requirements.

The following section presents and justifies the simulation platform used for this research. The remaining of this chapter is organised as follows: section 4.1 presents the simulation platform used for this research, section 4.2 presents the simulation studies structure for this research, section 4.3 presents the simulation results, section 4.4 presents the discussion of the results, and finally, section 4.5 concludes the chapter.

4.1 CloudSim Simulator

Since our targeted system is an Infrastructure as a Service (IaaS) configuration which is a cloud computing environment that is supposed to create a view of unbounded computing resources to users, we find it very important to evaluate the proposed approach on a large-scale experiment on real infrastructure. However, it is obviously very difficult to conduct repeatable experiments on such large-scale cloud infrastructure, hence, we opt to simulate the configuration under study in order to validate and test the efficiency of the proposed system model. For that, we have chosen the CloudSim toolkit [26] as the simulation platform for this research.

The CloudSim toolkit is considered a modern simulation framework aimed at Cloud computing environments [8] and it is widely used. The CloudSim toolkit permits the modeling of virtualized environments and supports on-demand resource provisioning and management. It also facilitates simulating energy-aware models; this makes it more useful

for our simulation than alternative simulation toolkits such as GangSim [63] and SimGrid [64].

In order to explain the technical details of the CloudSim simulator, we find it useful to draw the implementation of a typical cloud-based datacentre which illustrates the concept of simulated cloud datacentres. Figure 4.1 presents the layered architecture of a typical cloud-based datacentre [26].

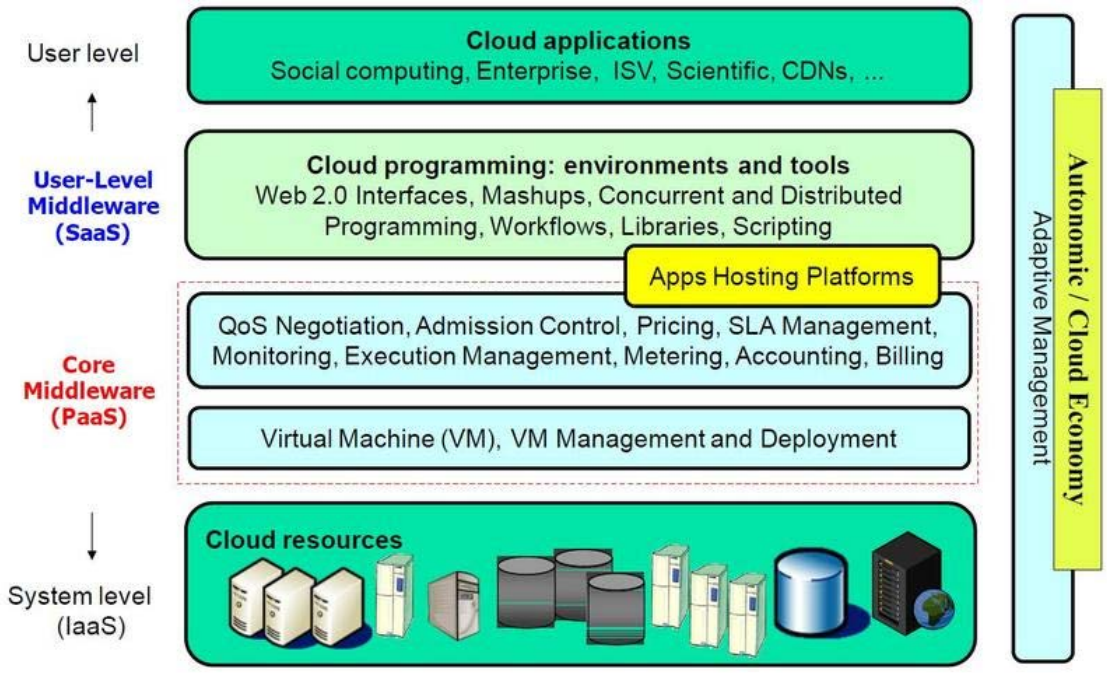


FIGURE 4.1 LAYERED ARCHITECTURE OF CLOUD-BASED DATACENTER [19]

As shown in Figure 4.1, the layered architecture of a cloud-based datacentre comprises four main layers: System Level (IaaS) which is the lowest layer in the architecture, Core Middleware (PaaS) that comes above the lowest, User Level Middleware (SaaS), and finally the top layer in the architecture is the User Level. The IaaS layer in the architecture involves massive physical resources such as storage and servers, these physical resources are the main power consumers in the datacentre. The PaaS layer includes virtualized

services and toolkits which allow sharing of their capacity among virtual instances of servers. PaaS describes the development, deployment and run-time services of cloud applications. In [65], the authors say: “If IaaS is the individual nuts and bolts that power the cloud, and SaaS is the user operating it, then PaaS is the link between both of them”. Typically, the PaaS facilitates the deployment of applications, application development, testing, and also supports the building, testing and hosting of Web applications. In a cloud datacentre, the PaaS layer provides access to operating systems and associated services, and a way to deploy applications to the cloud using programming languages and tools supported by the provider [66].

The SaaS layer is the highest layer in the cloud-based datacentre architecture. It is responsible for the operation of the emerging cloud applications such as social networking, business applications, gaming portals, and content delivery. The SaaS layer includes the software frameworks which helps developers in creating rich, cost effecting user-interfaces for browser-based applications. Each application in the SaaS layer requires QoS according to users’ interaction patterns such as online or offline.

4.1.1 CloudSim Architecture

CloudSim is a simulation platform used to simulate cloud computing environments. The CloudSim toolkit supports modeling and creation of one of more virtual machines (VMs) on a simulated node of a datacenter, and the creation of jobs and their mapping to suitable VMs. CloudSim also allows the simulation of multiple datacenters hence it enables a study on federation and associated policies for migration of VMs for reliability and automatic scaling of applications [26].

As shown in figure 4.2, the layered implementation of the CloudSim simulator consists of three main layers: SimJava, GridSim, CloudSim, and the User Code layer.

SimJava is the lowest layer in the CloudSim architecture; it is considered the discrete event simulation engine which implements the core functionalities required for higher-level simulation functions such as queuing and processing of events, creation of system components, communication among components, and management of the simulation clock.

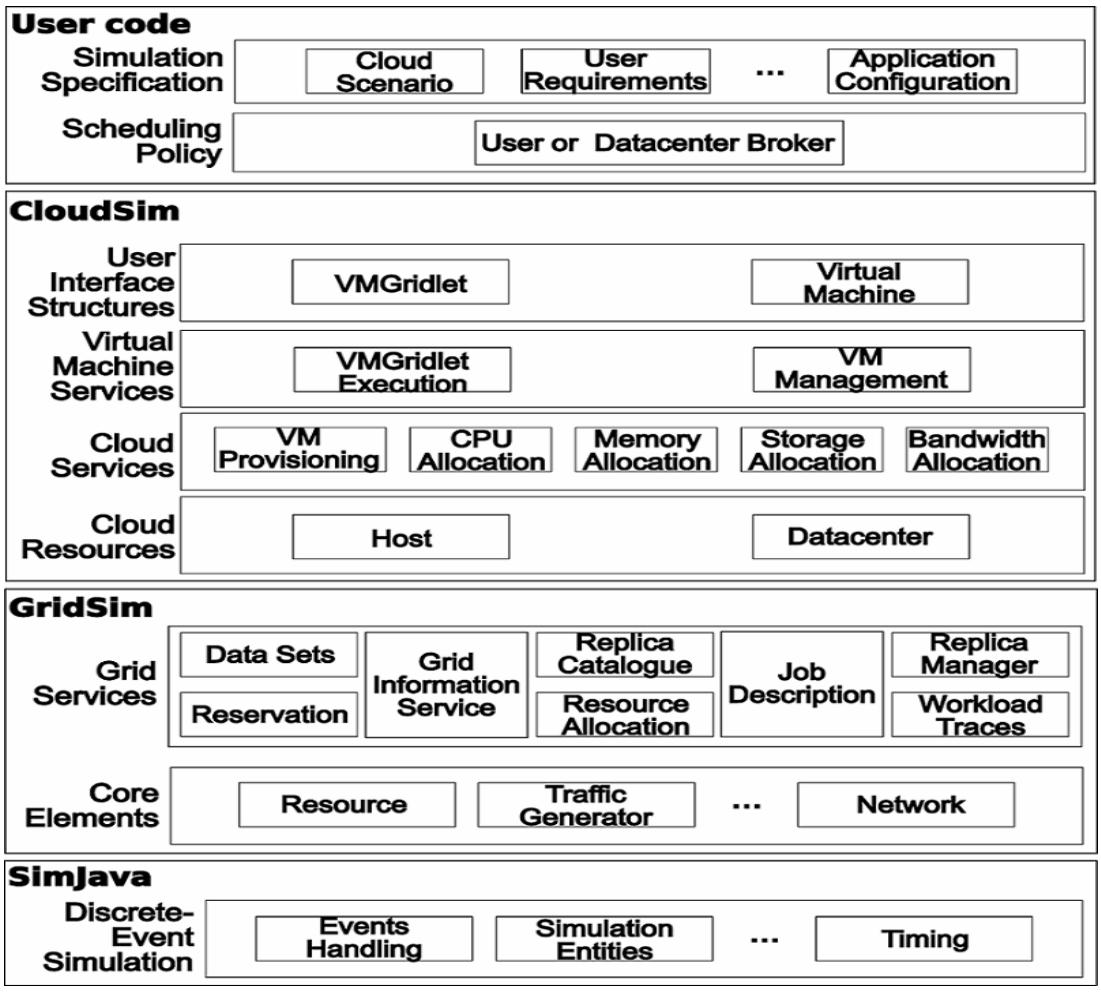


FIGURE 4.2 LAYERED ARCHITECTURE OF THE CLOUDSIM SIMULATOR [26]

The second lowest layer in the CloudSim architecture is the **GridSim** layer. This layer contains the libraries that implement the GridSim toolkit. The GridSim toolkit supports high level software components for the modeling of multiple Grid infrastructures, including networks and associated traffic profiles, as well as the fundamental Grid components such as the resources, data sets, workload traces and information services.

The **CloudSim** Layer is implemented above the GridSim layer as a programming extension of the core functionalities to the exposed GridSim layer. It provides novel support for modeling and simulation of virtualized cloud-based datacenters environments such as dedicated management interfaces for virtual machines, memory, storage, and bandwidth. The CloudSim layer is responsible for handling the management of application execution and dynamic monitoring. It is capable of concurrent instantiation and transparently manages a large scale cloud infrastructure consisting of thousands of system components.

The top layer in the CloudSim architecture is the **User Code** layer; it is the top-most layer in the simulation stack which exposes configuration related functionalities for hosts, applications and broker scheduling policies. At the User Code layer, a cloud application developer can generate a mix of user request distributions, application configuration and cloud availability scenarios. This helps developers to simulate and test scenarios based on the custom configurations that are already supported within the CloudSim.

4.1.2 Essential Entities in CloudSim

In order to understand how CloudSim can be used to simulate cloud-based environments, we find it useful to explain the essential entities involved in the simulation processes. In

CloudSim, each entity is represented as a class in the Java language. More information on CloudSim entities/classes can be found in [26].

a. Datacenter

The Datacenter entity comprises a set of hosts. It is responsible for managing operations and processes related to virtual machines such as provisioning. The Datacenter entity is considered as the IaaS provider; it receives requests from brokers and creates virtual machines accordingly.

b. Datacenter Broker

The datacenter broker is the entity that represents the user. Its responsibility is twofold: first, it modifies the mechanism for submitting virtual machines provisioning requests to datacenters; second, it modifies the mechanism for submitting tasks to virtual machines.

c. Host

The Host entity in CloudSim is responsible for the management of virtual machines such as creating and destroying and updating tasks to them. Hosts in CloudSim can also be a virtual machine and are associated to a datacenter.

d. Virtual Machine (VM)

The Virtual machine entity represents the implementation of a machine that is able to execute applications. VMs work like physical machines, each VM divides the resources received from the host among tasks running on it.

e. Cloudlet

The Cloudlet entity is basically a task. It is a Java class that represents the complexity of an application in terms of its computational requirements. Cloudlets in CloudSim are

managed by scheduling policies implemented in the datacenter broker entity/class mentioned earlier.

4.2 Simulation Studies Structure

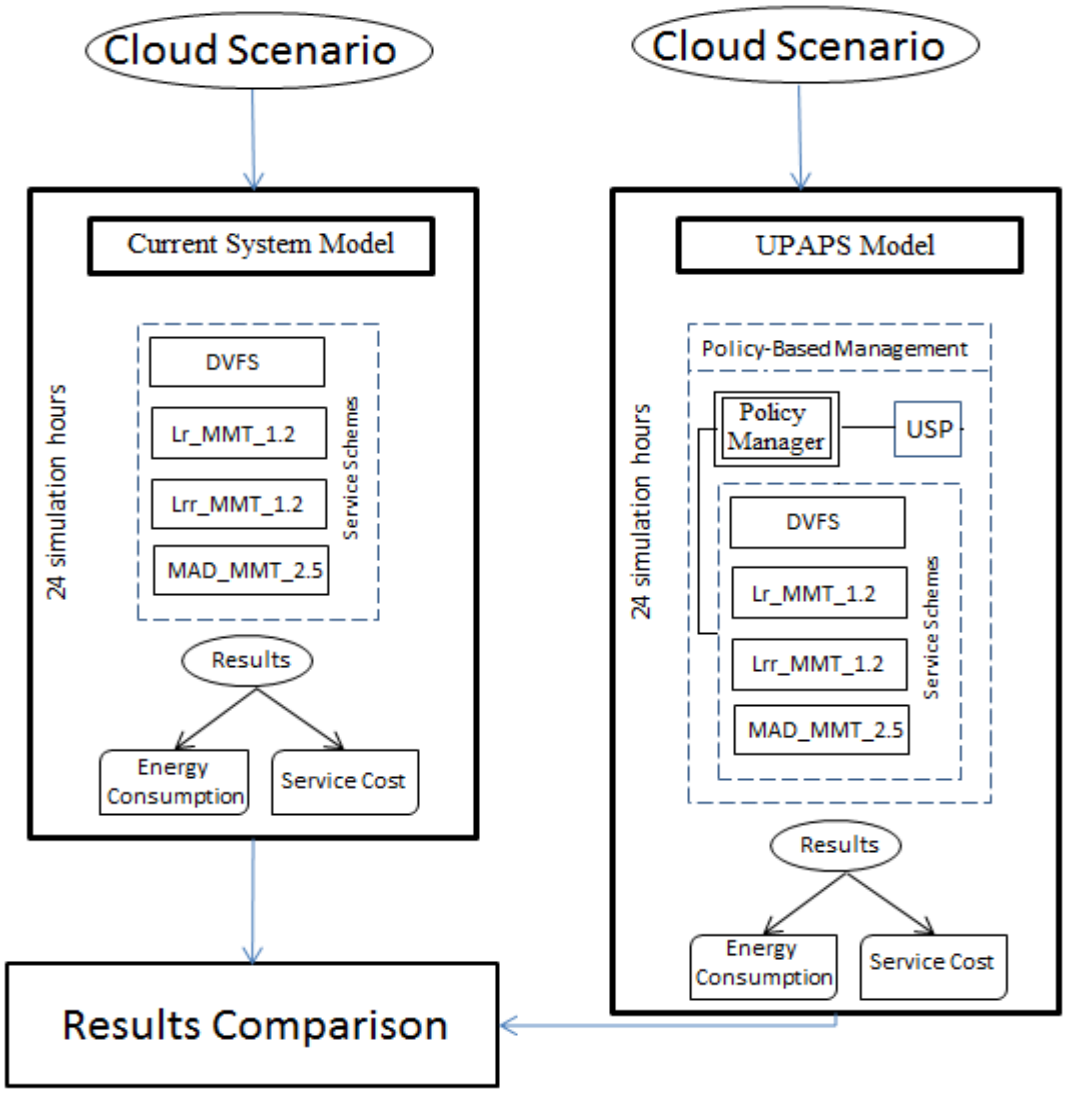


FIGURE 4.3 SIMULATION STUDIES STRUCTURE

In order to validate our proposed approach which aims at overcoming the issue of inefficiency of energy consumption in cloud computing, our simulation studies involved

two main stages in which the current system model and our proposed system model are both simulated. The results of both simulation stages are recorded for the goal of comparing them against each other to find out the efficiency of the proposed system model in terms of energy consumption. For the validation of our proposed approach, we opt to use two different scenarios in which each one has different cloud user requirements in terms of needed service schemes. For accurate results and fair results comparison in this study, we have chosen to run the simulation for all clients' scenarios under similar network hardware configurations; this assures fairness and accuracy in the final comparison and validation of results.

4.3 Network/Hardware Configurations

For accurate results and successful system validation in this research, we intend to run our simulation of cloud scenarios on similar network hardware configurations. This enables to accurately compare results of both simulation runs (current system model vs. proposed system model) and therefore allows accurate validation.

Similar to the simulation scenario built in [8], our simulation scenarios for this research involved running a data center that comprises 800 heterogeneous physical nodes, half of them are HP ProLiant ML110 G4 servers, and the other half are HP ProLiant ML110 G5 servers. The CPU frequency for each server is mapped onto MIPS ratings: 1860 MIPS for each core of the HP ProLiant ML110 G5 servers and 2660 MIPS for each core of the HP ProLiant ML110 G5 servers.

The network bandwidth for each server is 1 GB. Virtual Machines (VMs) characteristics correspond to Amazon EC2 instance types [67] with the only exception that all the VMs are single-core.

4.4 Cloud Scenarios and Simulation Studies

For this research, in order to validate our proposed approach, we opt to design two cloud scenarios in which two clients have different requirements in terms of SLA. The cloud scenarios in this research are simulated according to the same network configurations discussed in the previous section of this chapter. The main goal of creating cloud scenarios is to test our proposed approach against the current system model. Both cloud scenarios are dedicated to one assumed cloud provider scenario, in which there are many service schemes offered to clients. Service schemes in our cloud provider scenario vary according to the SLA provided and therefore priced to end consumer (clients). Our users scenarios are based on assumption about users' IT service preference with particular consideration to their daily usage of the cloud service. Each user scenario is simulated according to both: current system model and proposed system model. This allows testing the efficiency of the proposed system model and validating it by comparing the results obtained for both systems. For the rest of the chapter, the two user scenarios are denoted as Scenario 1, and Scenario 2.

4.4.1 Cloud Provider's Scenario

For this research, we assume a scenario of a cloud provider that provides services to clients based on their IT service preferences with particular consideration to their daily usage of the cloud service as mentioned earlier. In our cloud provider scenario, there are four available service schemes offered by the cloud provider in which each service scheme is

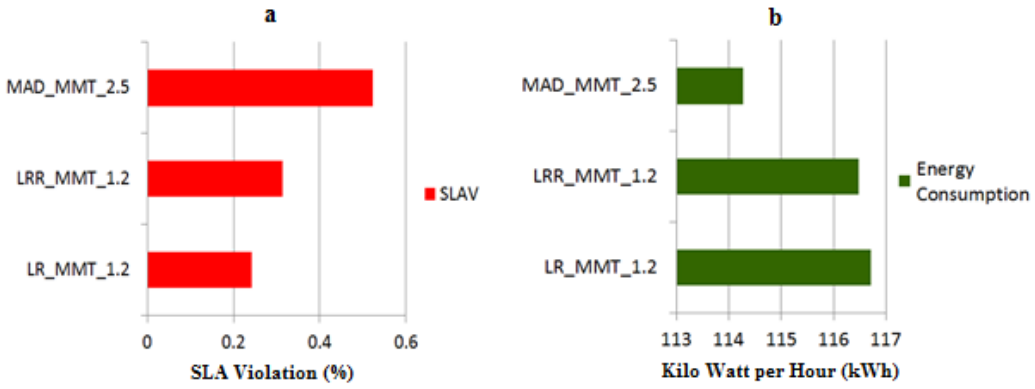


FIGURE 4.4 SIMULATION RESULTS OF THE SELECTED HEURISTIC

priced differently according to the QoS (SLA) offered in each scheme.

In order to generate our service schemes for this research, we opt to simulate the chosen heuristic algorithms discussed in section (3.2) to find their efficiency levels in terms of energy consumption and SLAV. For that, we have run a full day simulation on each heuristic algorithm and results were recorded to be put on the comparison bench. Results of this simulation are presented Figures 4.4 (a) and 4.4 (b).

As seen in figure (4.4) (a), the dynamic consolidation algorithm (DVFS) outperformed the heuristic algorithms in terms of the SLAV metric. The energy consumption rate caused by

DVFS in one full simulation day was 803.91 kilo watt per hour (kWh) which is much higher than the energy consumed by the heuristic algorithms. However, results indicated zero SLA violations caused by DVFS while the heuristic algorithms produced violations due to their use of aggressive VMs consolidations.

In figure (4.4) (b), the power consumption rates varied among algorithms. The LR_MMT_1.2 algorithm produced the highest energy consumption rate which was 116.71 (kWh), followed by LRR_MMT_1.2 which produced 116.48 (kWh), finally, MAD_MMT_1.2 produced an energy consumption rate of 114.27 (kWh). However, the efficiency of each heuristic algorithm does not only rely on the power consumption, but on the SLAV metric.

Comparing the results shown in figure 4.4 (a) and 4.4 (b), it is clear there is an inverse relationship between the power consumption and the SLAV metric. This is justified by aggressive consolidation of virtual machines that causes less energy consumption. For instance, MAD_MMT_2.5 with SLAV of 0.524% appears to produce the best results in terms of energy consumption, while in fact, the reason behind this energy efficiency is the high violation to the SLA due to its aggressive consolidation processes.

Based on the results in both figures 4.4 (a) and 4.4 (b), the heuristic algorithms can be categorized according to the SLAV (violations) they cause. None of these algorithms can be efficient when used on their own. The SLAV metric is considered very important in the field of cloud computing, even though all heuristic algorithms contribute towards saving energy, they are still not efficient to be applied practically due to the SLAV that they cause. Moreover, although the DVFS algorithm does not cause violation to the SLA, the energy

consumption rate that it causes is considered relatively high according to the target consumption rate in our proposed approach. Nevertheless, applying the DVFS algorithm on its own would put cloud users in a position to pay for high QoS all the time even if they don't actually need it.

For our cloud scenario in this research, we opt to categorize service schemes according to the SLAV metric that each heuristic algorithm produces.

These service schemes are: service scheme "A" which provides the highest SLA due to its use of the DVFS algorithm, service scheme "B" which has the second highest SLA due to the LR-MMT-1.2 algorithm, service scheme "C" which provides lower SLA which is generated using the LRR-MMT-1.2 algorithm, and finally service scheme "D" which offers the lowest SLA due to MAD-MMT-1.2 algorithm.

Table 4.1 presents service schemes prices according to our cloud provider scenario. Also, we here assume the cost of 1 Kilo Watts of power is equal to $NZ 0.17; which is part of the cloud operating costs.

Service Scheme	Price (Per Hour)
Service A	$NZ 0.34
Service B	$NZ 0.27
Service C	$NZ 0.20
Service D	$NZ 0.13

TABLE 4.1 PRICE PLAN FOR SERVICE SCHEMES

4.4.2 Cloud Users' Scenarios

For this thesis, as mentioned earlier, we assume two cloud user scenarios in which each user has different requirements of service schemes according to their needs. The first user is a bank and the second user is a university office.

4.4.2.1 Scenario 1 (Bank)

The first user scenario represents a bank branch that requires the highest available quality of service during business hours, and the second highest quality of service outside business hours. The bank branch manager specifies a great need of high internet connection speed during business hours due to the high workload on the bank intranet, while having a lower need to that outside their business hours due to less workload on the intranet of the bank.

A. Current System Model

According to the current system model, only one policy (algorithm) can be enforced. In order to achieve **0.0 SLAV**, service scheme "A" is the proper service scheme to be offered as it satisfies the need of the bank by providing the highest available SLA level, this is due to the need of high quality of service during business hours hence lower quality of service can violate the SLA between the cloud provider and the bank during business hours. Service scheme "A" is generated using DVFS algorithm, we here run our simulation for one full simulation day (24 hours). Table 4.2 presents the results of our simulation for one full simulation day in terms of energy consumption as part of the cloud running costs, as well as the service costs that the bank has to pay as the price of the service within the context of the pay as you go.

The energy consumption rate as per CloudSim simulator for DVFS algorithm is 52.98 Kilo Watt per Hour (kWh), and the price of the service as per our assumption in table (4.1) equals to NZ$0.34.

The total energy consumption in one full day using DVFS is per CloudSim simulator is calculated as follows:

803.91 * 24 = **19293.84 Kw**

While the price of service scheme "A" in one full day is calculated as follows:

0.34 * 24 = **8.16 NZ$**

Simulation results of one full day of DVFS is presented in the following table

Energy Consumption	Service Price
19293.84 Kw	$8.16

TABLE 4.2 DVFS SIMULATION RESULTS (24 HOURS)

Due to the fact that the current system is only using one dedicated scheme which does not allow flexibly switching policies/service schemes according to the dynamic user needs, the service scheme "A" is provided to the bank throughout the full working day. This makes the bank pay for such service during outside business hours without an actual need to it.

B. The UPAPS system model

In our proposed system model, the adoption of Policy-Based Network Management (PBNM) facilitates more elasticity in managing service schemes according to the actual need of them. According to the bank scenario, the bank has the facility of choosing service

schemes from a pool of service schemes according to what is needed with consideration to time windows. In this scenario, the bank requires the service scheme that provides service with the highest available highest available quality of service. The bank requirements can be summarized as the following:

From 8am – 4:59 pm, service scheme “A”
From 5pm – 7:59 am, service scheme “B”

The bank’s requirements can be satisfied by our proposed system model by designing the bank USP according to them.

Figure 4.5 presents the bank’s USP as per our proposed system model. This USP is used by the cloud provider in order to set their service schemes schedule according to the required services and time windows. Moreover, in the proposed system model, users can also be part of the management for their own service schemes by modifying their service schemes using the user interface which is connected to the policy repository discussed in section (2.4.3).

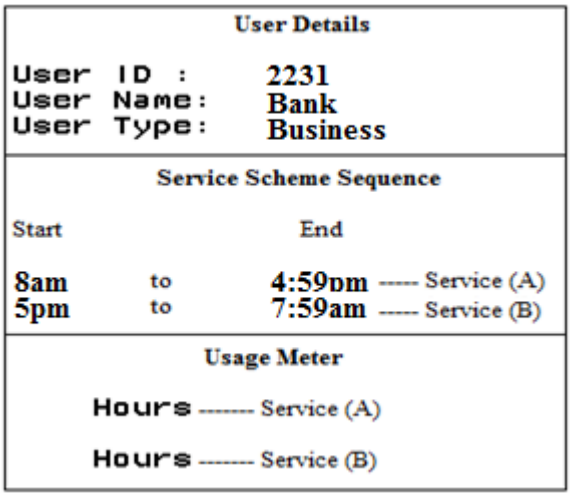


FIGURE 4.5 USP FOR THE BANK

As seen in figure 4.5, service schemes can be modified and set according to time windows as per the bank needs. The first part of the USP includes user details such as ID, name, and user type, while the second part involves the service schemes schedule in which the bank itself (user) can modify it and set it according to its needs. The third part involves the usage meter in which each service scheme is measured hourly and priced accordingly. This means, USP allows the bank to set up its service schemes schedule according to the required service as well as eliminating the costs of the ICT service provided by the cloud service provider.

Figure 4.6 presents the energy consumption rate in one full simulation day according to the USP of the bank.

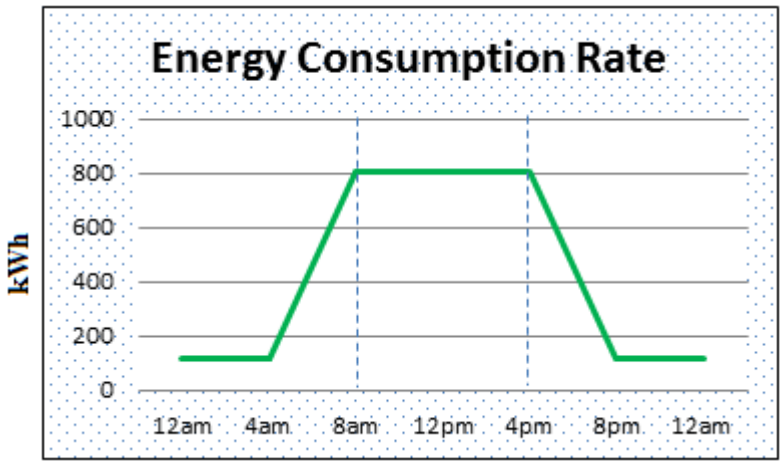


Figure 4.6 ENERGY CONSUMPTION In 1 FULL BUSINESS DAY

As seen in figure 4.6, during business hours, service scheme “A” is applied, which is generated using the DVFS algorithm, while outside business hours, service scheme “B” is applied.

The total energy consumed in 1 full business day according to figure 4.6 can be calculated as follows:

12am to 8am service scheme “B” 8 Hours (116.71 * 8) = **933.68 Kw**

8am to 4pm service scheme “A” 8 Hours (803.91 * 8) = **6431.28 Kw**

4pm to 12am service scheme “B” 8 Hours (116.71 * 8) = **933.68 Kw**

Total Energy consumed in one full business day = 8298 Kw

The price that the bank has to pay for 1 full business day service as per its USP is also calculated as follows:

16 Hours of Service Scheme “B” + **8 Hours** of Service Scheme “A”

16 * NZ$ 0.27 + 8 * NZ$ 0.34

Total cost for one full business day = NZ$ 7.04

4.4.2.1.1 Results Discussion

Results of both simulation stages (current system model and proposed UPAPS system model) indicate to significant reduction in the energy consumption using our proposed system model, as well as significant saving in terms of service costs that the bank is entitled of paying to the cloud provider. Figure 4.7 (a) presents the total reduction of energy consumption by our proposed system model, while figure 4.7 (b) presents the cost reduction of cloud service using the proposed system model.

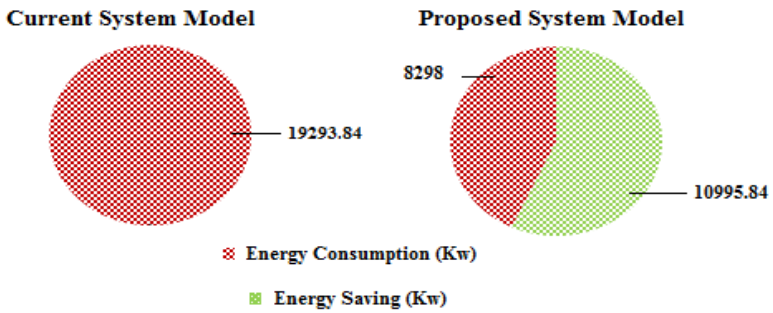


FIGURE 4.7 (a) ENERGY CONSUMPTION REDUCTION

As seen in figure 4.7 (a), the results of simulations indicates to a significant energy consumption reduction in our proposed system model. The total energy consumed in the current system model equals to 19293.84 Kw in one business day, while according to the proposed system model, the energy consumed is reduced to 8298 Kw. This proves the efficiency of our proposed system model (UPAPS) in terms of energy consumption as the total energy saved is (19293.84 – 8298 = 10995.84 Kw) which is 57% of the total energy consumed in the current system model is saved without violating the SLA that the bank requires.

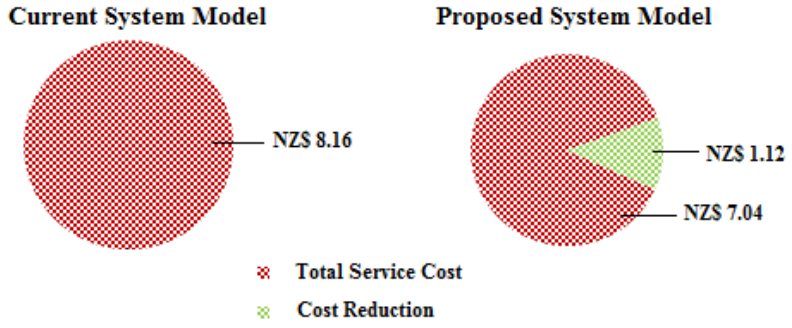


FIGURE 4.7 (b) SERVICE COST REDUCTION (NZ$)

Moreover, according to the simulation of the bank scenario, results also point to significant cost reductions of the cloud service provided to the bank. As seen in figure 4.7 (b), the cost

of the service for one full day is reduced by (8.16 – 7.04 = 1.12NZ$) which is around 13% of the total cost in the current system model hence; this also proves the cost efficiency of our proposed system model in comparison to the current one.

4.4.2.2 Scenario 2 (University Office/department)

Similarly, the second scenario represents a university office manager who requires cloud services for their office ICT. The manager of the university office decides to reduce the costs of ICT by choosing service schemes according to their demand. The service schemes required by the office manager are not fully dependent on the opening hours of the branch, but rather on the working time and workload (number of computers that are sharing the network bandwidth in the office at each particular time window of the day). The requirements of the office manager are more complex that the bank in our previous scenarios. These requirements include 3 service schemes which are scheduled according to both; working time and workload.

The manager of the university office requires the highest available quality of service during business hours, due to the need for fast internet connection and voice and video communications that usually take place during working hours. During outside business hours, due to the nature of the office work, there is always a possibility that some of the staff members are doing overtime work in order to accomplish tasks when needed. This means that the workload of the office network is unpredictable during outside business hours. For that, the manager of the office requires scheduling service schemes according to the number of running computers on the office network. In other words, when the number of computers running is more than 3, the workload is considered high, and if the number of

running computers on the office network is less than 3, the workload is considered low and so on. The number of computers influence the speed of the internet as each computer shares the internet bandwidth with the other running computers on the office network. The following represents the office requirements in terms of service schemes.

A. Current system model

According to the current system model, in order to satisfy the university office manager's requirements, and due to the need for service scheme "A" during business hours and possible after business hours when workload is considered high, the simulation results will not differ from the simulation results of the previous scenario (the bank), because service scheme "A" is generated using the DVFS algorithm which consumes energy at the rate of 903.91 Kilo watt per hour, which means in one full day, the total energy consumption can be calculated as follows

803.91 * 24 = **19293.84** Kw

However, the current system model does not support such requirements because only one service scheme can be provided throughout the day and therefore no complexity of requirements can be handled.

B. UPAPS system model

In order to satisfy the user's requirements, these requirements have to be included in the USP of the user as a set of policies. This set of policies plays the key role in the process of scheduling the service schemes provided to the user (university office). This is because of the unpredictable workload that the office network might encounter after business hours.

According to the requirements of the university office manager, the service scheme required during business hours is service scheme “A”, while after business hours, there is a need to create a set of policies that can automatically handle the scheduling and the switching processes of the service schemes without violating the quality of service required. This can be done by adopting the policy-based network management framework. Policies can be switched from one to another depending on the workload metric decided by the office manager which is the number of running computers on the network.

The following presents the set of policies which are to be included in the USP of the office:

During Business Hours ⟶ Service Scheme “A”

Outside Business Hours

1. If number of working computers $= 0$ ⟶ Service Scheme “D”
2. If number of working computers $> 0 \leq 3$ ⟶ Service Scheme “B”
3. If number of working computers < 3 ⟶ Service Scheme “A”

Figure 4.8 presents the user’s USP in which the set of policies above is included in order to satisfy the requirements of the office owner.

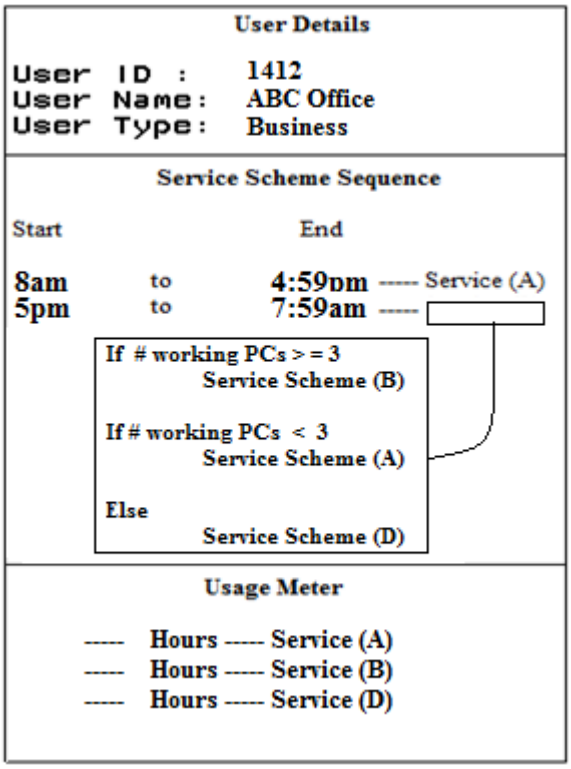


FIGURE 4.8 USP OF THE UNIVERSITY OFFICE

As seen in figure 4.8, service schemes are provided to the office network according to both metrics; time and workload. During business hours, only one service scheme is provided which is service scheme "A", while after business hours, there are three different service schemes which can be provided according to the workload of the network at any particular time outside business hours. This reflects the flexibility that the proposed approach, along with the policy-based network management framework provides. Moreover, the advantage of the policy-based network management framework also resides in the simplification of the management of the cloud network as switching from a service scheme to another does not require human effort once policies are set and enforced in the system.

4.4.2.2.1 Policy Conflicts and Resolve

Policy conflicts occur due to the satisfaction of conditions of two or more policies in the system at the same time. The adoption of policy-based management framework can often encounter policy conflicts that can be a barrier towards the adoption of such management framework. In our office scenario, policies were set up and designed to be enforced according to two metrics: time and network workload therefore, there is no occurrence of policy conflicts, however, on larger scales, there is always a potential for policy conflicts that take place due to more requirements to be satisfied and higher demand on greater number of metrics in various applications or users types scenarios. For example, let us assume that the design of our policies set in the office scenario was different in a way that Service Scheme “B” was to be provided during outside business hours unless there is a workload on the network, and other service schemes are to be provided according to the size of the workload on the network. In this case, in order to overcome the issue of policy conflicts, we would set up a unique priority value for each policy with its service scheme. For example, the policy for service scheme “A” has the priority of “3”, policy for service scheme “B” has the priority of “2” and policy for service scheme “D” has the priority of “1”. According to that, outside business hours, Service “D” is supposed to be provided, but when the workload metric starts to increase, the priority therefore would be for the other policy that has higher priority value such as policy “B”. Similarly, if the workload metric increases to satisfy the condition of the policy for the service scheme “A”, the priority for the policy for service scheme “A” is higher therefore, service “A” is to be enforced and so on.

Prioritizing policies in the policy-based network management framework helps towards avoiding policy conflicts and therefore obtaining the best of what such a management framework can offer.

For our simulation, we opt to further create a sub-scenario derived from the office scenario for the goal of validating our proposed system model in terms of energy consumption and costs. For that, we assume the following:

During outside business hours, the workload on the office network was as follows:

From 5pm to 8pm, three computers were running due to 3 staff members who were performing overtime work in the office. From 8pm to 9pm, the manager of the office was doing some work in the office using one computer of the office network. After 9pm, the office network was not occupied and none of the office computers were running.

According to the mentioned sub-scenario, the workload of the office network after business hours according to the USP of the office can be summarized as the following:

1. 3 hours of service scheme “A”
2. 1 hour of service scheme “B”
3. 11 hours of Service scheme “D”

According to the simulation runs on the assumed sub-scenario of the office branch, the total energy consumption for each of the above time windows is calculated as the following

3 hours of service scheme “A” ⟶ 3 * (103.09 kWh) ⟶ 309.27 Kw

1 hour of service scheme “B” ⟶ 1* (116.71 kWh) ⟶ 116.71 Kw

11 hours of service scheme “D” ⟶ 11* (55.08 kWh) ⟶ 605.88Kw

Total Energy consumed = 309.27 + 116.71 + 605.88 = **1031.86** Kw

The total cost of the cloud scenario according to our sub-scenario is calculated as the following:

3 hours of service scheme “A” ⟶ 3 * (0.34 NZ$) ⟶ 1.02 $NZ

1 hour of service scheme “B” ⟶ 1* (0.27 NZ$) ⟶ 0.27 $NZ

11 hours of service scheme “D” ⟶ 11* (0.13 NZ$) ⟶ 1.43 $NZ

The total cost of the cloud service = 1.02 + 0.27 + 1.43 = **2.72** $NZ

Simulation results of one full day according to our proposed system model is presented in the following table

Energy Consumption	Service Price
1031.86 Kw	**NZ$ 2.72**

TABLE 4.3 SIMULATION RESULTS ACCORDING TO PROPOSED SYSTEM MODEL

4.4.2.2.2 Results Discussion

Results of both simulation stages (current system model and UPAPS system model) indicate a significant reduction in the energy consumption using our proposed system model, as well as significant saving in terms of service costs that the university office is required to pay to the cloud provider. Figure 4.9 (a) presents the total reduction of energy consumption by our proposed system model, while figure 4.9 (b) presents the cost reduction of cloud service using the proposed system model.

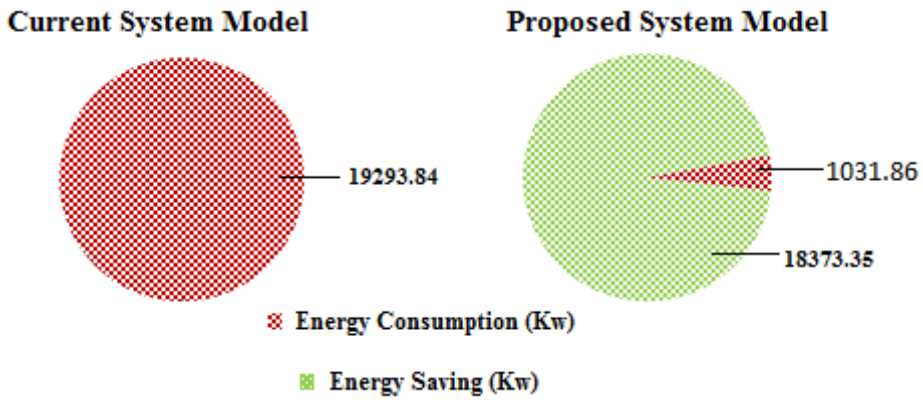


FIGURE 4.9 (a) TOTAL ENERGY CONSUMPTION

As seen in figure 4.9 (a), there is a significant reduction in the total energy consumed by running the cloud scenario according to the user's USP under the proposed system in comparison to the current system model. The total energy reduction due to the UPAPS model is (19293.84 – 1031.86 = 18261.98 Kw) which makes around 95% of the total energy consumed according to the current system model. This indicates better efficiency in terms of energy consumption without violation to the quality of service required by the university office.

Moreover, the cost of service appears to be significantly reduced according to the proposed system model as seen in figure 4.9 (b). The total reduction of the cost of service according to the UPAPS model is (8.16 – 2.72 = 5.44 NZ$), this indicates significant reduction of service cost which is around 66% of the total costs of service according to the current system model.

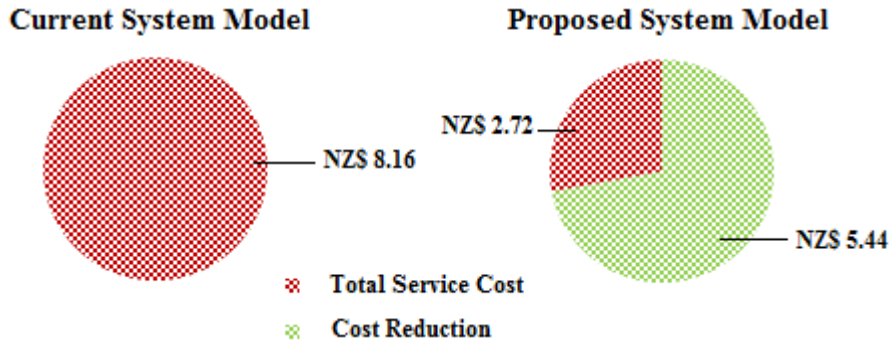


FIGURE 4.9 (b) SERVICE COST REDUCTION

4.5 Summary

The proposed system model involves the adoption of the policy-based network management framework concept in which a group of heuristic algorithms can be managed towards the reduction of the energy consumption in the field of cloud computing without violation to the quality of service required. The proposed UPAPS system model and cloud architecture which involves the two proposed architectural components i.e., USP and UPAPS, work together towards better management of cloud networks as well as providing more means of elasticity in the field of cloud computing under the umbrella of the pay-as-you-go model.

In this chapter, we first draw two network scenarios which involve one cloud service provider who provides a group of service schemes to clients, and two different users who demand different service schemes according to their own requirements in terms of time slots of the day. Secondly, for the validation of our proposed system model and architecture concept, we have run extensive simulations on both scenarios under both systems: the current system model and the proposed system model (UPAPS).

We have found that the UPAPS system model causes less energy consumption in comparison to the current system model, due to the ability of switching service schemes which are generated using heuristic algorithms in which the energy consumption rates vary from one to another according to the technical details of each algorithm in terms of virtual machine migrations on the cloud provider side. Moreover, we have also found that the UPAPS system model decreases the cost of service due to the less energy consumed and therefore less operating costs of the cloud.

In conclusion to the simulation results and findings, UPAPS model is found to be more energy efficient than the current system model, this is because the UPAPS model is more flexible and works explicitly according to users' needs. The reason behind the more energy efficiency in the UPAPS model lies on the accuracy of computing resources allocation in which energy consumption reduction is the direct consequence to that.

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

The main purpose of this thesis was to provide an in-depth understanding of the fundamentals of the cloud computing energy consumption issue. We were striving to provide a contribution for new knowledge and solution for the issue of insufficiency of energy consumption in the field of cloud computing by using the virtualization approach and virtual machine migration together with a policy-based network management framework.

5.1 Contributions

Inspired from the existing heuristic algorithms which contribute towards the reduction of energy consumption in virtualized cloud-based environment in the current cloud system model, we have identified its weaknesses, and tried to tackle these problems by proposing a new cloud system model and architecture which builds on what other researchers have achieved in the field of virtualization and migration, as well as network management techniques. The contributions of this work advance the field of cloud computing in two ways; first, achieving sufficient trade-offs between energy consumption in datacenters without aggressively affecting the quality of service delivered. Second, proposing a new cloud-based architecture concept that offers a range of service level agreements (SLAs) via a policy-based network management framework.

The first contribution of this work involved proposing a new system model that can handle and manage a group of heuristic algorithms proposed by other researchers. Several heuristic algorithms have been built and proposed by other researchers; however, none of these heuristic algorithms can satisfy the need of a sufficient trade-off between energy consumption and quality of service on its own. The proposed UPAPS framework has proved ability in managing a group of these heuristic algorithms by employing our proposed UPAPS algorithm. The UPAPS algorithm together with the USP components are significant contribution towards the efficient management of heuristic algorithms and have achieved the desired trade-off between energy consumption and quality of service delivered. This is considered as a contribution towards helping with the development of a strong and competitive cloud computing industry using green design concepts.

The second contribution of this work is that our proposed cloud system model allows obtaining more benefits of what cloud computing can potentially offer on the basis of pay-as-you-go elasticity, because users according to our proposed approach can design their service schemes and pay for what they exactly need on one hand, and cloud operators can reduce their operating costs to make it more beneficial in terms of business on the other hand. We have presented and evaluated our approach towards better cloud computing design. The simulation results have shown that the adoption of a policy-based network management framework works effectively towards reducing the energy consumption in cloud datacenters. Results also showed that the policy switching approach is efficient for both cloud service providers and cloud users.

5.2 Future Work

Considering the work covered in this thesis within a constraint in the limited time period and the development of the future implementation of cloud system model, we find it useful to highlight some future areas to be further investigated as cloud computing energy efficiency remains an open field for future research and investigations

The heuristic algorithms used in this thesis were limited to a few algorithms which have been developed and proposed by other researchers. For our future work, we aim at designing more energy-efficient algorithms to employ in our proposed system model towards better trade-offs between energy efficiency and quality of service as well as more flexibility in the cloud computing field in terms of service schemes.

Moreover, the work conducted throughout this thesis involved proposing and validating a new policy-based system model and architecture for cloud computing, however, the actual implementation of such system remains as a future target to achieve due to the limited hardware resources provided by the university. Moreover, our proposed approach towards overcoming the issue of energy consumption involved the adoption of the PBNM; this is also another milestone for our future research to implement and find out how such management framework (PBNM) can be built and adjusted in order to self-manage a cloud-based network.

References

[1] T. B. Winans and J. S. Brown, "A collection of working papers," *Cloud Computing* , p. 2, 2009.

[2] E. A. Fischer and P. M. Figliola, "Overview and Issues for Implementation of the Federal Cloud Computing Initiative: Implications for Federal Information Technology Reform Management," *Congressional Research Service*, 2013.

[3] P. Mell and G. Timothy, *The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology*, 2011.

[4] M. Mishra, A. Das, P. Kulkarni and A. Sahoo, "Dynamic Resource Management Using Virtual Machine Migrations," in *Cloud Computing: Networking and communication challenges*, 2012.

[5] S.-i. Kuribayashi, "Reducing Total Power Consumption Method in Cloud Computing Environments," *International Journal of Computer Networks & Communications*, pp. 69-84, 2012.

[6] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Elsevier: Future Generation Computer Systems*, pp. 599-616, 2008.

[7] J. M. Pedersen, M. Riaz, J. C. Junior, B. Dubalski, D. Ledzinski and A. Patel, "Assessing Measurements of QoS for global Cloud Computing Services," *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing* , pp. 682-689, 2011.

[8] A. Beloglazov and R. 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* , pp. 1-24, 2011.

[9] Greenpeace, "Cloud Computing and its Contribution to Climate Change," Greenpeace International , Amsterdam, 2010.

[10] J. G. Koomey, "A report by Analytics Press, completed at the request of The New York Times," Stanford University, USA , 2011.

[11] S. Rivoire, M. A. Shah, P. Ranganathan and C. Kozyrakis, "Joulesort: a Balanced Energy-Efficiency Benchmark," *ACM SIGMOD International Conference on Management of Data,*, 2007.

[12] S. Srikantaiah, A. Kansal and F. Zhao, "Energy Aware Consolidation for Cloud Computing," in *HotPower* , 2008.

[13] L. A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," *Computer* , pp. 33-37, 2007.

[14] H. Jin, H. Liu, X. Liao, L. Hu and P. Li, "Live Migration of Virtual Machine Based on Full-System Trace and Replay," in *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI 2005), USENIX*, Boston, MA, USA, 2005.

[15] L. MacVittie, "Cloud Computing: It's the destination, not the journey that is important," 03 November 2008. [Online]. Available: http://www.google.co.nz/imgres?imgurl=https://devcentral.f5.com/weblogs/images/devcentral_f5_com/weblogs/macvittie/WindowsLiveWriter/ExistentialTechnologyCloudComputingandSO_45AA/cloudcomputing-f5_2.jpg&imgrefurl=https://devcentral.f5.com/articles/cloud-c. [Accessed 15 September 2013].

[16] D. C. Plummer, T. J. Bittman, T. Austin, D. W. Cearley and D. M. Smith, "Cloud Computing: Defining and Describing an Emerging," in *Gartner*, 2008.

[17] B. Williams, "The Economics of Cloud Computing: An Overview For Decision Makers," in *Cisco Press*, 2012.

[18] N. Mirzaei, "Cloud Computing," *Indiana University*, 2009.

[19] Dialogic, "Introduction to Cloud Computing," *Dialogic*, 2010.

[20] B. Kepes, "Understanding the cloud computing stack: SaaS, PaaS, IaaS," *Rackspace Hosting, Diversity* , pp. 1-17, 2011.

[21] KloudPros, "Types of Cloud Deployments," [Online]. Available: http://www.kloudpros.com/primer-cloud-computing/types-of-cloud-deployments/. [Accessed 09 November 2013].

[22] J. Leonard, Systems engineering fundamentals, Virginia: DIANE Publishing, 2001,

pp. 117-124.

[23] S. Schlesinger, "Terminology for model credibility," *Simulation* , pp. 103-104, 1979.

[24] D. K. Pace, "Modeling and Simulation Verification and Validation," *Johns hopkins Apl Technical Digest,* pp. 163-172, 2004.

[25] A. Maria, "Intoruduction to modeling and simulation," in *Proceedings of the 1997 Winter Simulation Conference*, Binghamton, 1997.

[26] R. R. Rodrigo N. Calheiros, A. Beloglazov, C. A. F. D. Rose and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Practice and Experience (SPE),* pp. 23-50, 2011.

[27] S. T. March, "Design Science in the Information Systems. Discpline: An Introduction to the special issue on design science research," *Miss Quarterly* , pp. 725-730, 2008.

[28] D. K. Kumar, G. Rao and G. Rao, "Cloud Computing: An Analysis of Its Challenges & Security Issues," *International Journal of Computer Science and Network (IJCSN),* 2012.

[29] L. Willcocks, W. Venters and E. A. Whitley, "Meeting the challenges of cloud computing," *Accenture* , 2011.

[30] Q. Zhang, L. Cheng and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *J Internet Serv Appl (2010),* pp. 7-18, 2010.

[31] L. Zyga, "How energy-efficient is cloud computing?," 8 October 2010. [Online]. Available: http://phys.org/news205737760.html. [Accessed 11 August 2012].

[32] W. H. Gong Chen, J. Liu, S. Nath, L. Rigas, L. Xiao and F. Zhao, "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services," *5th USENIX Symposium on Networked Systems Design and Implementation,* pp. 337-350, 2008.

[33] K. C. Feeney, D. Lewis and V. P. Wade, "Policy Based Management for Internet Communities," in *Fifth IEEE International Workshop on Policies for Distributed Systems and Networks*, Ireland, 2004.

[34] R. Chadha, G. Lapiotis and S. Wrighi, "Policy- Based Networking," in *IEEE*

Network, 2002.

[35] T. Hamada, P. Czezowski and T. Chujo, "Policy-based Management for Enterprise and Carrier IP Networking," *Fujitso,* pp. 128-139, 2000.

[36] M. Dam, G. Karlsson, B. S. Firozabadi and R. Stadler, "A Research Agenda for Distributed Policy-based Management," *International Journal of P2P Network Trends and Technology,* pp. 116-122, 2013.

[37] R. Berto-Monleon, E. Casini, R. v. Engelshoven, R. Goode, K.-D. Tuchs and T. Halmai, "Specification of a Policy Based Network Management Architecture," *The 2011 Military Communications Conference, Track 3, Cyber Security and Network Operation ,* pp. 1393-1398, 2011.

[38] D. Agrawal, K.-W. Lee and J. Lobo, "Policy-Based Management of Networked Computing Systems," *IEEE Communications Magazine,* pp. 69-75, 2005.

[39] D. Agrawal, K.-W. Lee and J. Lobo, "Policy-based Management of Networked Computing Systems," *IEEE Communications Magazine,* pp. 69 - 75, 2005.

[40] D. C. Verma, "Simplifying Network Administration Using Policy-Based Management," *IBM Thomas J Watson Research Cente, IEEE Network ,* pp. 20-26, 2002.

[41] RIPE, "Internet Engineering Task Force," 10 August 2012. [Online]. Available: http://www.ripe.net/internet-coordination/internet-governance/internet-technical-community/ietf.

[42] M. Davies, C. Clark and D. Legare, "Proceedings of the Twenty-Fourth Internet Engineering Task Force," in *Twenty-Fourth Internet Engineering Task Force ,* Cambridge, 1992.

[43] J. Rich and J. Hill, "How to Do Capacity Planning," *TeamQuest,* 2013.

[44] Y. J. Song, V. Ramasubramanian and E. G. Sirer, "Optimal Resource Utilization in Content Distribution Networks," *Dept. of Computer Science, Cornell University, Ithaca,* 2006.

[45] A. Balchunas, "Overview of IPSEC," 2007.

[46] H. L. McKinley, "SANS Institute InfoSec Reading Room," *Sans Institute ,* 2003.

[47] J. Follows and D. Straeten, "Application-Driven Networking: Concepts and Architecture for Policy-Based Systems," *International Technical Support Organization, IBM,* 1999.

[48] D. Armstrong and K. Djemame, "Towards Quality of Service in the Cloud," in *Proceedings of the 25th UK Performance Engineering Workshop*, UK, 2009.

[49] V. Firoiu, J.-Y. L. Boudec, D. Towsley and Z.-L. Zhang, "Theories and Models for Internet Quality of Service," in *Proceedings of the IEEE*, 2002.

[50] Cisco, "Quality of Service Networking," in *Internetworking Technologies Handbook*, Indianpolis, Cisco Press, 2004, pp. 1-31.

[51] Szigeti, Tim and C. Hattingh, End-to-End QoS Network Design: Quality of Service in LANs, WANs and VPNs, Cisco Press, 2004.

[52] J. Tiso, "Enterprise QoS Solution Reference Network Design Guide," in *Designing Cisco Network Service Architectures (ARCH) Foundation Learning Guide*, USA, Cisco Press, 2005.

[53] N. Kokash, "An introduction to heuristic algorithms," *Department of Informatics and Telecommunications, University of Trento, Italy,* 2012.

[54] F. Greco, Travelling Salesman Problem, Vienna, Austria: In-Teh, 2008.

[55] G. Winter, B. Galvn, S. Alonso and B. n. G. alez, "Solving Economic and Environmental Optimal Control of Dumping of Sewage with a Flexible and Parallel Evolutionary Computation," *Institute of Entelligent Systems and Numerical Applications in Engineering , Evolutionary Computation and Application Division, University of Las Palmas De gran Canaria ,* pp. 244-247, 2000.

[56] W. S. Cleveland, "Robust Locally Weighted Regression and Smoothing Scatterplots," *Journal of Statistical Americal Association ,* pp. 829-836, 1979.

[57] Y. Cai, Q. Cui, P. Huss, S. Lu, L. Richardson and T. Yang, "T, F, AND CAUCHY," *Distribution Project: Group 1,* 2013.

[58] E. L. Sueur and G. Heiser, "Dynamic Voltage and Frequency Scaling: The Laws of Diminishing Returns," in *HotPower'10 Proceedings of the 2010 international conference on Power aware computing and systems*, USA, 2010.

[59] V. Spiliopoulos, S. Kaxiras and G. Keramidas, "Green Governors: A Framework for Continuously Adaptive DVFS," in *Green Computing Conference and Workshops (IGCC), 2011 International* , Orlando, FL, 2011.

[60] E. Vittoz and J. Fellarath, "CMOS Analog Integrated Circuits Based on Weak Inversion Operation," *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, pp. 224-231, 1977.

[61] Y. Wang and X. Wang, "Power Optimization with Performance Assurance for Multi-tier Applications in," *University of Tennessee, Knoxville*, pp. 1-8, 2010.

[62] L. A. Lymberopoulos, "An Adaptive Policy Based Framework for Network Management," *Department of Computing, Imperial College London, University of London*, 2004.

[63] M. Bsoul, I. Phillips and C. Hinde, "MICOSim: A simulator for modelling economic scheduling in Grid computing," *World Academy of Science, Engineering and Technology*, pp. 1298-1301, 2012.

[64] A. Legrand, M. Quinson, H. Casanova and K. Fujiwara, "The SimGrid Project," *Simulation and Deployment of Distributed Applications*, 2006.

[65] J. Rundle, "Waterstechnology," 02 July 2012. [Online]. Available: http://www.waterstechnology.com/waters/feature/2188624/layer-cake-rise-platform-service.

[66] IBM, "Cloud computing: Introduction to Platform as a Service," 7 October 2011. [Online]. Available: http://www.ibm.com/developerworks/training/kp/cl-kp-cloudpaas/.

[67] Amazon, "Amazon EC2 instance types," 2013. [Online]. Available: http://aws.amazon.com/ec2/instance-types/. [Accessed 9 May 2013].

[68] N. Kasar, "Cloud Models Redefined in 3 ways," 2013. [Online]. Available: http://www.google.co.nz/imgres?imgurl=http://2.bp.blogspot.com/-VY8yZeMiw5o/UZs74JvlLKI/AAAAAAAAAIo/y48AyOgzzR4/s1600/Cloud-Deployment-Models.png&imgrefurl=http://www.thetechaddas.com/2013/06/cloud-models-redefined-in-3-ways.html&usg=__oaX5XIw5lhUw0Fnq9_9. [Accessed 06 November 2013].

Glossary

CPU	Central Processing Unit
DMTF	Distributed Management Task Force
DVFS	Dynamic Voltage Frequency Scaling (Algorithm)
ICT	Information and Communication Technology
IaaS	Infrastructure as a Service
IETF	Internet Engineering Task Force
IQR	Inter Quartile Range (Algorithm)
Kwh	Kilo Watt per Hour
LR	Local Regression (*Algorithm*)
LRR	Local Regression Robust (*Algorithm*)
MAD	Median Absolute Deviation (*Algorithm*)
MMT	Minimum Migration Time (*Algorithm*)
MC	Maximum Correlation (*Algorithm*)
MIPS	Million Instructions per Second
NPA	None Power Aware Algorithm (*Algorithm*)
PaaS	Platform as a Service
PBNM	Policy-Based Network Management
PMAC	Policy Management for Autonomic Computing
PDP	Policy Decision Point

PEP	Policy Enforcement Point
PDM	Performance Degradation due to Migration
PBNM	Policy-Based Network Management
QoS	Quality of Service
RS	Random Selection (*Algorithms*)
RAM	Random Access Memory
SLA	Service Level Agreement
SaaS	Software as a Service
SLATAH	SLA Violation Time per Active Host
SLAV	Service Level Agreement Violation
TCA	Total Cost Acquisition
THR	Static Threshold VM allocation policy (*Algorithm*)
UPAPS	User Profile Aware Policy Switching
USP	User Service Profile
VM	Virtual Machine

APPENDIX A: CLOUDSIM TOOLKIT

The CloudSim is open source Java-based simulation toolkit described as an extensible simulation framework that allows seamless modeling, simulation, and experimentation of merging Cloud computing infrastructure and application services. The main advantages obtained from using the CloudSim toolkit are time effectiveness as it requires less effort and time to implement cloud-based application provisioning test environment, flexibility and applicability as developers can model and test the performance of their application services in heterogeneous cloud environment with little programming and deployment

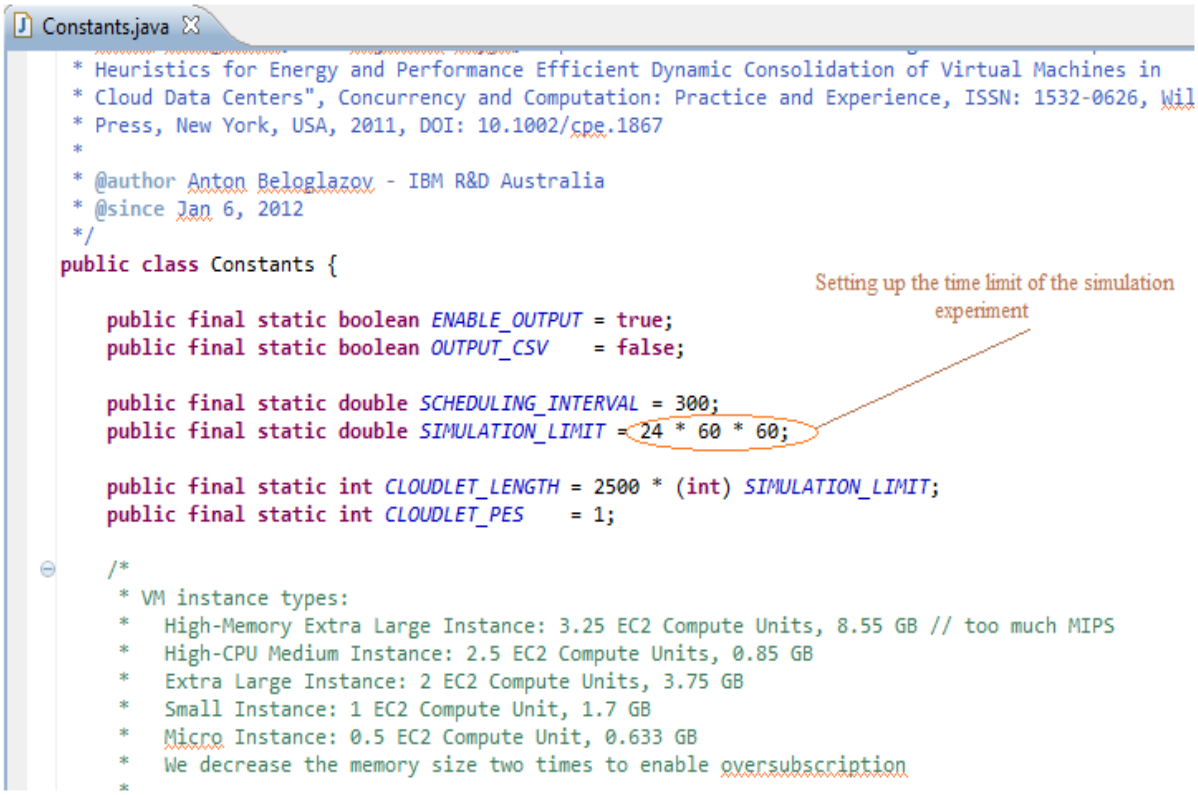


FIGURE A.1 CONFIGURATION SCREEN OF THE CLOUDSIM TOOLKIT

efforts. The CloudSim toolkit enables the simulation of Cloud networks based on required

configurations and constants. Figure A.1 presents a screenshot of the constants page configuration in the CloudSim where the specification of a cloud-environment experiment can be set or modified according to the intended goal of the experiment.

The circle points out the time limit in seconds of the simulation which is set to be 24 hours simulation.

Figure A.2 represents a screenshot of the constants page in the CloudSim toolkit which shows the configurable constants that are related to the network devices and other virtual machine-related specifications.

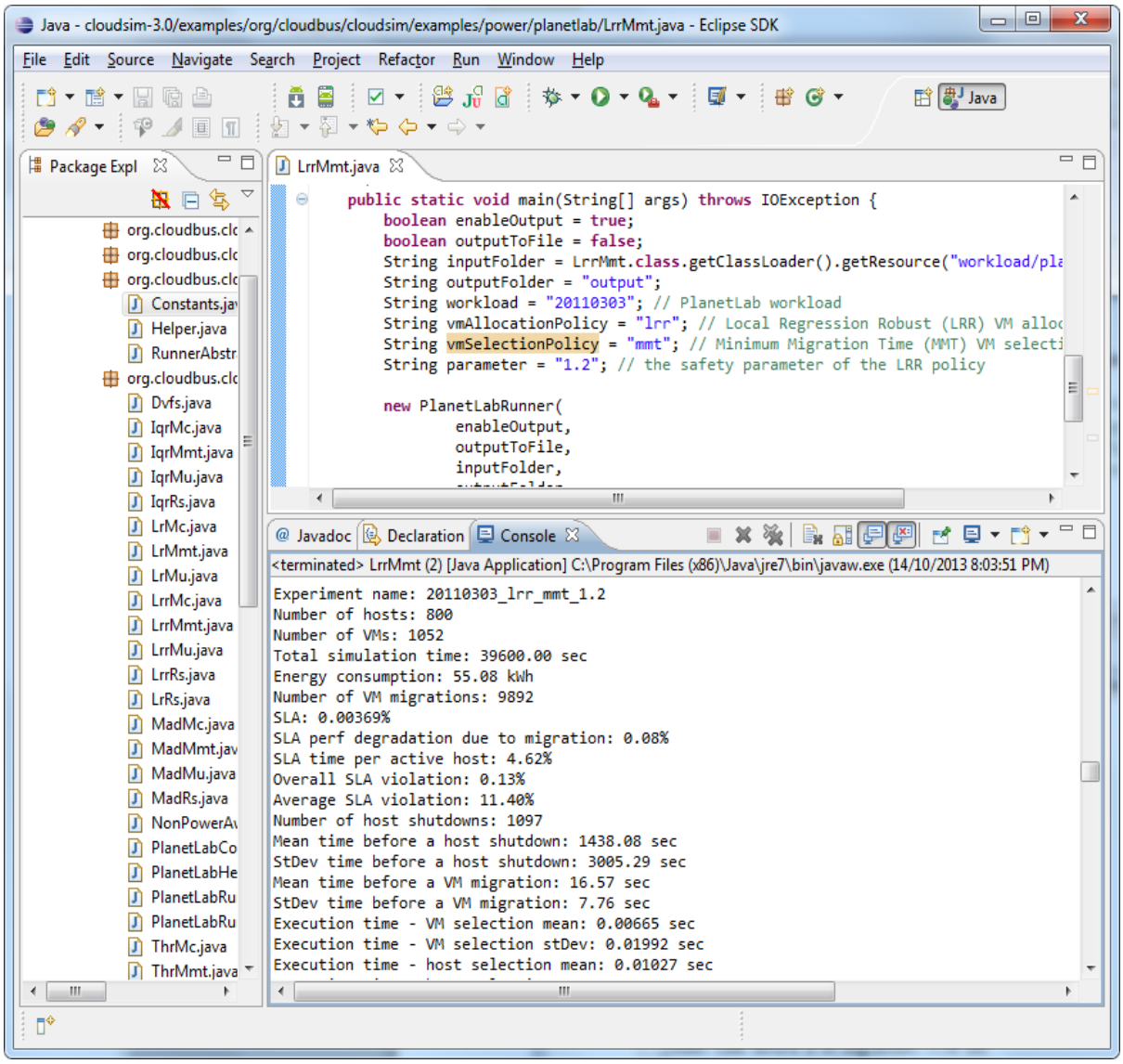
```
/*
 * VM instance types:
 *   High-Memory Extra Large Instance: 3.25 EC2 Compute Units, 8.55 GB // too much MIPS
 *   High-CPU Medium Instance: 2.5 EC2 Compute Units, 0.85 GB
 *   Extra Large Instance: 2 EC2 Compute Units, 3.75 GB
 *   Small Instance: 1 EC2 Compute Unit, 1.7 GB
 *   Micro Instance: 0.5 EC2 Compute Unit, 0.633 GB
 *   We decrease the memory size two times to enable oversubscription
 *
 */
public final static int VM_TYPES    = 4;
public final static int[] VM_MIPS   = { 2500, 2000, 1000, 500 };
public final static int[] VM_PES    = { 1, 1, 1, 1 };
public final static int[] VM_RAM    = { 870,  1740, 1740, 613 };
public final static int VM_BW       = 100000; // 100 Mbit/s
public final static int VM_SIZE     = 2500; // 2.5 GB

/*
 * Host types:
 *   HP ProLiant ML110 G4 (1 x [Xeon 3040 1860 MHz, 2 cores], 4GB)
 *   HP ProLiant ML110 G5 (1 x [Xeon 3075 2660 MHz, 2 cores], 4GB)
 *   We increase the memory size to enable over-subscription (x4)
 */
public final static int HOST_TYPES   = 2;
public final static int[] HOST_MIPS  = { 1860, 2660 };
public final static int[] HOST_PES   = { 2, 2 };
public final static int[] HOST_RAM   = { 4096, 4096 };
public final static int HOST_BW      = 1000000; // 1 Gbit/s
public final static int HOST_STORAGE = 1000000; // 1 GB
```

FIGURE A.2 CONFIGURATION SCREEN OF CLOUDSIM TOOLKIT

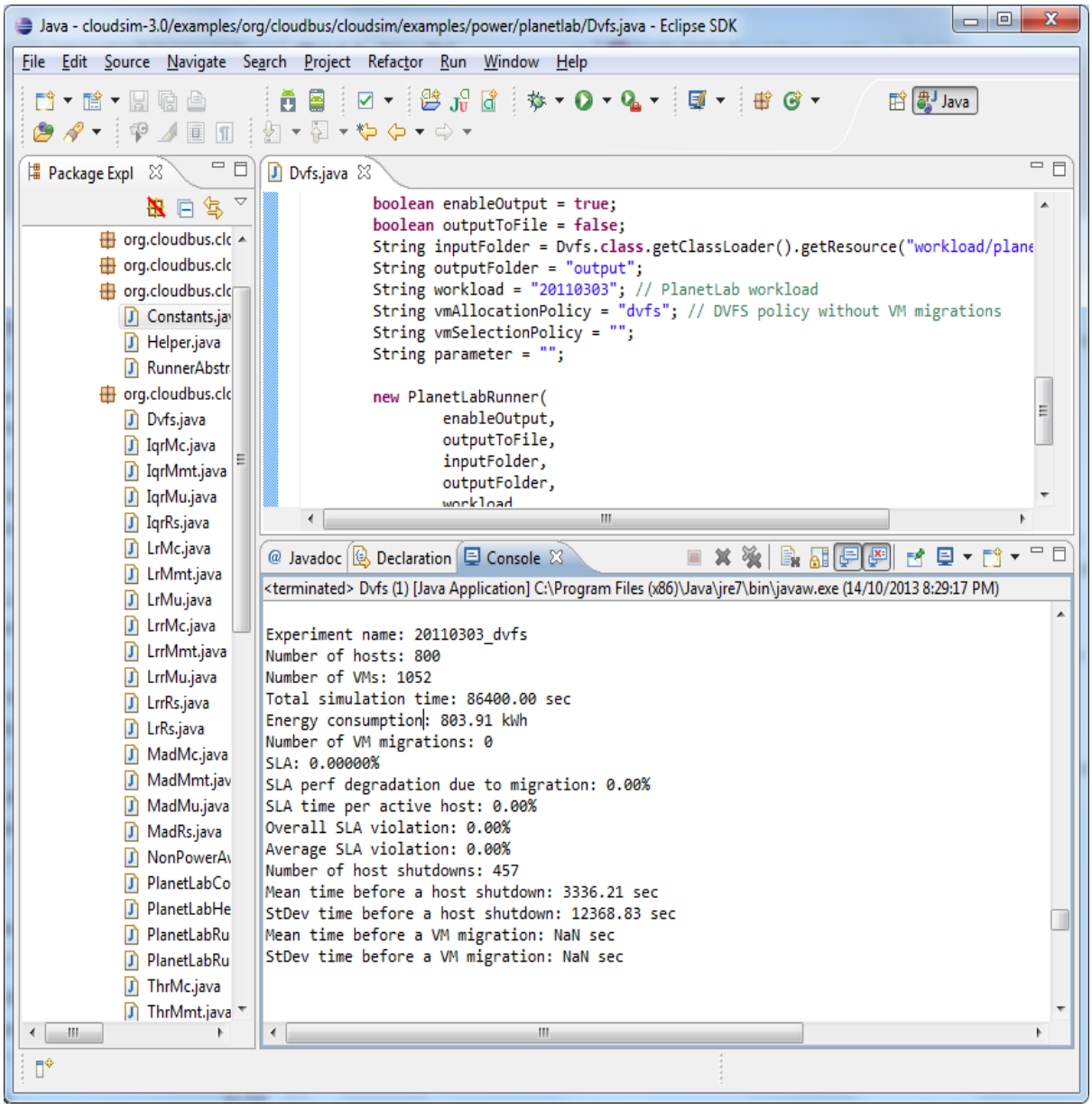
More information about the CloudSim toolkit is available in [26]. Moreover, the CloudSim toolkit as mentioned earlier is an open source toolkit and can be found using the following link: http://www.cloudbus.org/cloudsim/

APPENDIX B: SIMULATION RESULTS

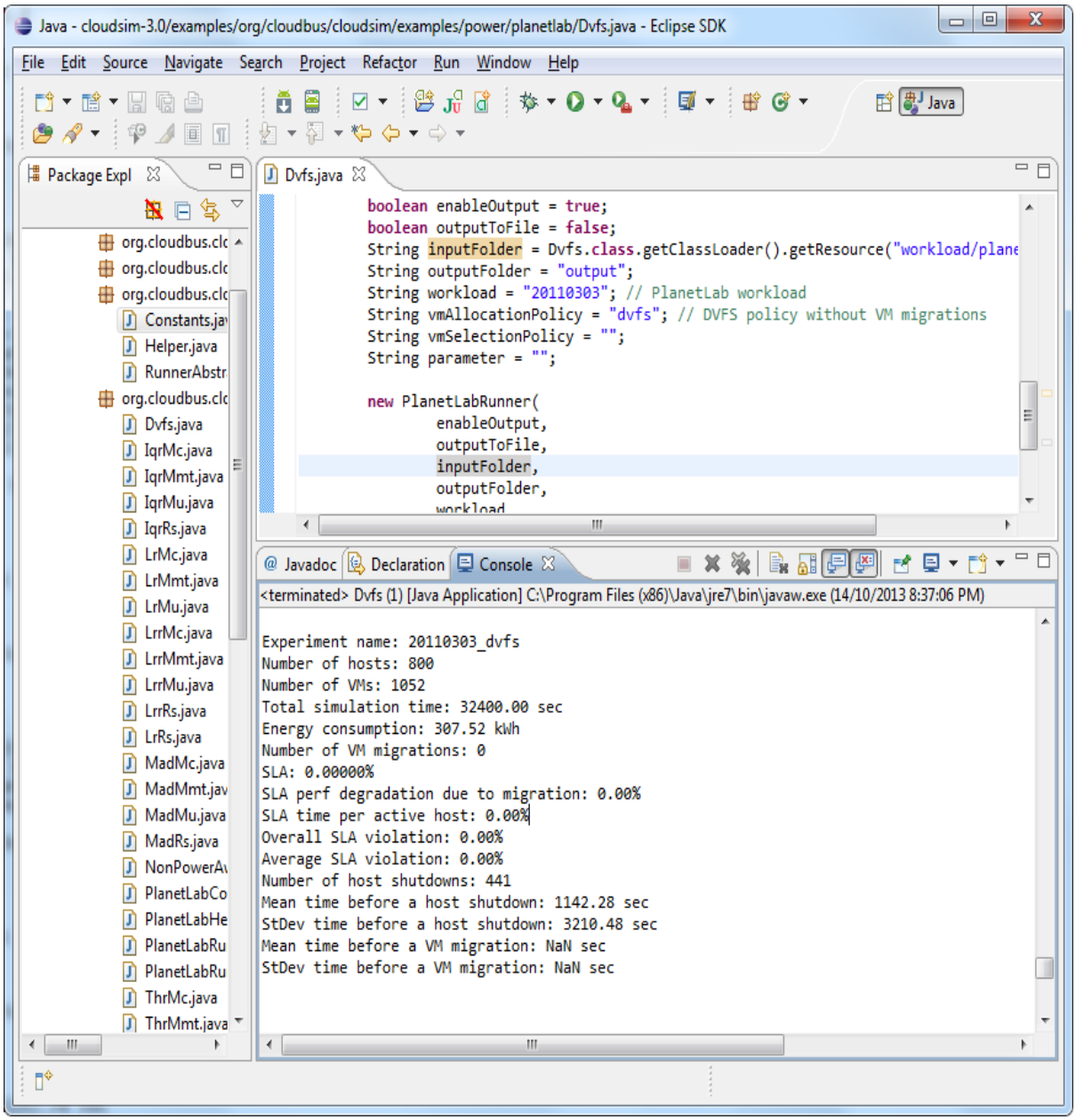
Simulation for Service Scheme "D" for 11 Hours



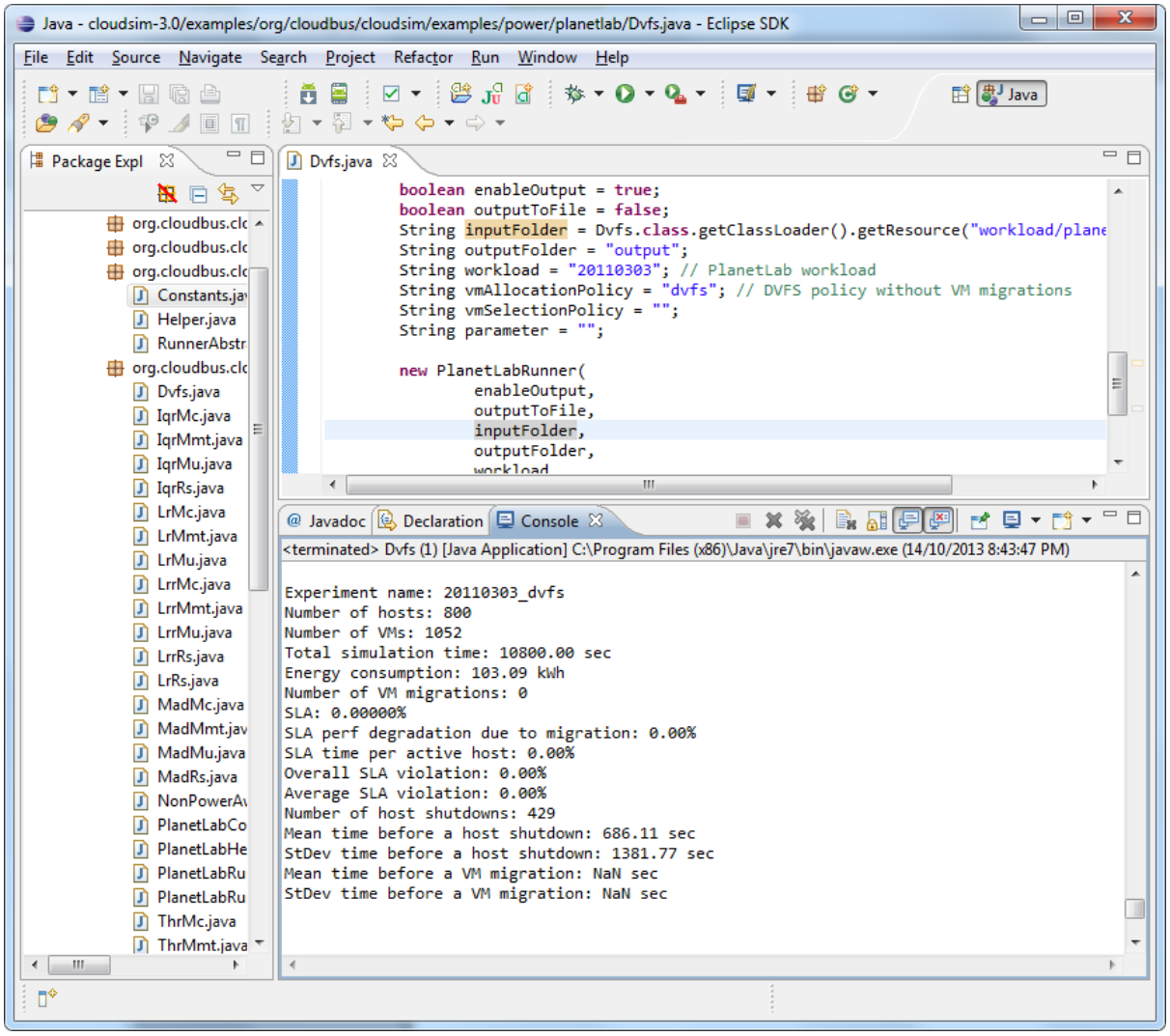
Simulation for Service Scheme “A” for 24 hours



Simulation for Service Scheme “A” for 9 hours



Simulation for Service Scheme “A” for 3 hours



Simulation for Service Scheme “B” for 15 Hours