Energy-Aware Profiling and Prediction Modelling of Virtual Machines in Cloud Computing Environments

By

Ibrahim Ali M Alzamil

Submitted in accordance with the requirements

for the degree of Doctor of Philosophy

The University of Leeds

School of Computing

May 2017

Declaration

The candidate confirms that the work submitted is his/her own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

I. Alzamil, K. Djemame, D. Armstrong, and R. Kavanagh. ***"Energy-Aware Profiling for Cloud Computing Environments"***. Proceedings of the 30th UK Performance Engineering Workshop, N. Thomas (Ed.), Newcastle, UK, September 2014. This paper is the candidate's own work. It was reviewed by the co-authors Karim Djemame, Django Armstrong and Richard Kavanagh. Content of this paper is included throughout the thesis and mainly in Chapter 4.

I. Alzamil, K. Djemame, D. Armstrong, and R. Kavanagh. ***"Energy-Aware Profiling for Cloud Computing Environments"***. Journal of Electronic Notes in Theoretical Computer Science (ENTCS), November 2015, 318, pp.91-108. Most of this paper's content is the candidate's own work. The co-author Django Armstrong provided the Leeds Cloud testbed technical support. The co-author Richard Kavanagh collaborated with the introduction of the energy modeller. The paper was reviewed by the co-author Karim Djemame. Its content is included throughout the thesis and mainly in Chapter 4.

I. Alzamil and K. Djemame. ***"Energy Prediction for Cloud Workload Patterns".*** Proceedings of the 13th International Conference on Economics of Grids, Clouds, Systems and Services (GECON'2016), Athens, Greece, September 2016. This paper is the candidate's own work. It was reviewed by the co-author Karim Djemame. Its content is included throughout the thesis and mainly in Chapters 5 and 6.

Acknowledgements

First and foremost, I thank *Allah* (God) for His blessings and granting me knowledge, patience, and health to complete this thesis.

I would like to thank my supervisor, Dr Karim Djemame, for his generous support and guidance throughout my PhD. This work would never have been possible without his critical feedback, encouragement and commitment, for which I am very grateful. Additionally, I would like to thank my examiners, Dr Gary Wills and Professor Jie Xu, for their excellent comments and feedback during the examination.

I also would like to acknowledge and thank Richard Kavanagh and Django Armstrong for their help and technical support during the experimental setup on the Cloud testbed. I would like to extend my thanks to all friends, colleagues, and to the members of the Distributed System and Services Research Group for their valuable discussions and support.

My great gratitude and appreciation goes to my parents, Ali and Hussah, for their enduring love and support throughout my life. Additionally, I express my gratefulness to my beloved wife, Tarfah, and son, Ryan, for being always there supportive and patient with this long journey. Also, my thanks go to my brothers and sisters for their help and encouragement.

Finally, I would like to thank Majmaah University for granting the scholarship to do my PhD studies in the UK.

Abstract

Cloud Computing has changed the way in which individuals and businesses use IT resources. Instead of buying their own IT resources, they can use the Cloud services offered by Cloud providers with reasonable costs based on a "*pay-per-use*" model. With the wide adoption of Cloud Computing, the costs for maintaining the Cloud infrastructure have become a vital issue for the providers, especially with the large amount of energy being consumed to operate these resources. Hence, the excessive use of energy consumption in Cloud infrastructures has become one of the major cost factors for Cloud providers. In order to reduce the energy consumption and enhance the energy efficiency of Cloud resources, proactive and reactive management tools are used with consideration of physical resources' energy consumption. However, these tools need to be supported with energy-awareness not only at the physical machine (PM) level but also at virtual machine (VM) level in order to make enhanced energy-aware decisions. As the VMs do not have physical interface, identifying the energy consumption at the VM level is difficult and not directly measured.

This thesis introduces an energy-aware Cloud system architecture that aims to enable energy-awareness at the deployment and operational levels of a Cloud environment. At the operational level, an energy-aware profiling model is introduced to identify energy consumption for heterogeneous and homogeneous VMs running on the same PM based on the size and CPU utilisation of each VM. At the deployment level, an energy-aware prediction framework is introduced to forecast future VMs' energy consumption. This framework first predicts the VMs' workload based on historical workload patterns, particularly static and periodic, using Auto-regressive Integrated Moving Average (ARIMA) model. The predicted

VM workload is then correlated to the physical resources within this framework in order to get the predicted VM energy consumption.

The evaluation of the proposed work on a real Cloud testbed reveals that the proposed energy-aware profiling model is capable of fairly attributing the physical energy consumption to homogeneous and heterogeneous VMs, therefore enabling energy-awareness at the VM level. Compared with actual results obtained in this testbed, the predicted results show that the proposed energy-aware prediction framework is capable of forecasting the energy consumption for the VMs with a good prediction accuracy for static and periodic Cloud application workload patterns.

The application of the proposed work is providing energy-awareness which can be used and incorporated by other reactive and proactive management tools to make enhanced energy-aware decisions and efficiently manage the Cloud resources. This can lead towards a reduction of energy consumption, and therefore lowering the cost of operational expenditure (OPEX) for Cloud providers and having less impact on the environment.

List of Abbreviations

AIC	Akaike Information Criterion
API	Application Programming Interface
ARIMA	Auto-Regressive Integrated Moving Average
BIC	Bayesian Information Criterion
CAEX	Capital Expenditure
CLI	Command-Line Interface
CNS	Consumption Near Sweet-Spot
CO2	Carbon Dioxide
CPU	Central Processing Unit
CUE	Carbon Usage Effectiveness
DPM	Dynamic Power Management
DVFS	Data Voltage and Frequency Scaling
EPU	Energy-aware Profiling Unit
EPREU	Energy-aware Prediction Unit
ERF	Energy Reuse Factor
GEC	Green Energy Coefficient
GUI	Graphical User Interface
HPC	High Performance Computing
IaaS	Infrastructure as a Service
ICT	Information and Communication Technology
I/O	Input/Output
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology

KVM	Kernel Based Virtual Machine
kW	Kilowatt
kWh	Kilowatt-Hour
LLC	Last-Level-Cache
LUT	Lookup Table
LXC	Linux Containers
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ME	Mean Error
MPE	Mean Percentage Error
MWU	Mann-Whitney U Test
NFS	Network File System
NIST	National Institute of Standards and Technology
OPEX	Operational Expenditure
OS	Operating System
OVF	Open Virtualised Format
PaaS	Platform as a Service
PM	Physical Machine
PSU	Power Supply Unit
PUE	Power Usage Effectiveness
QoS	Quality of Service
RMSE	Root Mean Squared Error
RMU	Resource Monitoring Unit
SaaS	Software as a Service
SEEP	Symbolic Execution and Energy Profiles
SLA	Service Level Agreement

SMEs	Small and Medium Enterprises
SVR	Support Vector Regression
UML	Unified Modelling Language
USB	Universal Serial Bus
VCPU	Virtual CPU
VIM	Virtual Infrastructure Manager
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMM	Virtual Machine Monitor or Manager
VPN	Virtual Private Network
W	Watt
Wh	Watt-Hour

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Research Motivation

The energy consumption in the information and communication technology (ICT) industry is an area of significant ecological and economic concern. According to Gartner [1], the ICT industry is responsible for about 2 percent of the global CO2 emission, which is similar to the amount caused by the aviation industry. Further, a data centre may consume about 100 times more energy compared to a typical office with the same size [2]. The emergent technology of Cloud Computing is considered as a way to help reduce the energy consumption of the ICT industry by moving some of the ICT infrastructure from decentralised environments at the end-users, like small and medium enterprises (SMEs), to a centralised and more controlled environment at the Cloud infrastructure providers. These providers make use of virtualisation in the management of ICT resources, which provides a simplified server administration, improved resource utilisation, and reduced IT costs.

However, the radical adoption of Cloud Computing technology has exposed a significant overhead in maintaining its infrastructure, which has become a major issue for the Cloud providers due to the associated high operational costs, such as energy consumption especially with the fluctuating electricity prices. Cloud Computing infrastructures consist of large computing resources that consume a large amount of energy in order to operate. Also, the excessive use of energy in Cloud infrastructures leads to more heat dissipated, which requires more cooling resources in order to avoid hot spots and service performance degradation; and these cooling resources would consume more energy as well. Therefore, Cloud providers consider energy consumption as one of the largest cost factors [3] with a substantial impact on the operational cost of a Cloud infrastructure [4], [5]. So, efficiently managing the energy consumed by the physical servers of the Cloud infrastructure can improve the overall energy consumption; in the sense that as the servers consume less power, the heat generated by these servers would be reduced, which would then reduce the need for cooling resources that consume a large amount of energy as well and result in more energy savings. Improving the energy efficiency of Cloud Computing has been an attractive research topic for both academia and industry as it has become increasingly significant for the future of the ICT [6].

The impact of energy consumption is not only dependent on the efficiency of the physical resources, but also on the policies deployed to manage these resources as well as the efficiency of the applications running on these resources [7]. Different methods have been used to efficiently manage the Cloud resources, all of which can be based on certain thresholds, called *reactive*, or based on prediction, called *proactive*. For example, once exceeding a certain threshold, 80% of CPU utilisation, some actions take place by reactive methods to increase resources and avoid service performance degradation. Proactive methods have the advantage of taking some actions at earlier stages to avoid reaching that threshold and maintain the expected performance. To enable such optimisation and the efficient design of Cloud applications, the software analysts and developers should be provided with energy information to support their programming decisions. Also, the deployment policies should incorporate energy information to make energy-efficient decisions when deploying these applications on the Cloud resources. As discussed in [8], having appropriate tools for energy monitoring and profiling is essential to support energy-awareness and contribute to energy optimisation in all layers of the Cloud stack. Further, tasks' workload information can help make efficient task placement strategies. As stated in [9], predicting the workload of a Virtual Machine (VM) is essential to make effective deployment strategies and energy efficient resource allocation decisions. Thus, managing the Cloud stack in all different levels and reducing the energy consumption has been an active area of research.

1.2 Research Context

Cloud Computing consists of a number of loosely coupled layers that can work in isolation and be supported by different providers [10]. In order to address energy efficiency through the whole stack of Cloud, energy information is needed to support various stakeholders, including Cloud application software analysts and developers as well as Cloud service providers.

- **The software analysts** need to incorporate the energy information when setting the applications requirements [11]. This requires specifying energy goals and designing models that make the applications adapt based on these goals.
- In order for **the software developers** to write energy efficient code, programming models that combine energy information are also needed [12], [13]. Programming models with energy-awareness can help the developers to make efficient programming decisions.
- **The Cloud service providers** also need to be supported with energy information in order to efficiently deploy the services with energy consideration. Additionally, energy information can help the Cloud service

providers for efficient placement and management of the VMs on the Cloud infrastructures [14].

Therefore, identifying and providing energy-awareness to these stakeholders is crucially significant to help them achieve their energy efficiency goals. The context of this research is applicable for a provider who owns the whole Cloud stack, meaning that they have control over all the Cloud layers; and the above stakeholders are part of one entity.

A Cloud application can run on one or many VMs, and these VMs can be hosted by one or many of Physical Machines (PMs). The energy consumption can be easily identified for the PMs with the use of hardware Watts meters. However, identifying the energy consumption at the VM level is difficult and not directly measured, and requires modelling the energy of the underlying PMs. Further, the energy consumption of an application can be identified by the total energy consumption of all the VMs on which this application is running.

1.3 Research Aim and Objectives

The overall aim of this research is to answer the following research questions:

- ***Q.1:*** How can energy-awareness be supported at the VM level in a Cloud system architecture?
- ***Q.2:*** How to fairly attribute the energy consumption to homogeneous and heterogeneous VMs running on the same PM?
- ***Q.3:*** How to proactively predict the energy consumption of the VMs prior to their deployment?

Thus, this research is aimed towards enabling energy-awareness of resource usage at virtual level in Cloud Computing environments, which contributes to overcome the challenge of identifying energy-awareness for the VMs. Also, this research aims to predict the future energy usage of the new requested VMs prior to deployment based on specific Cloud workload patterns. The outcome of this research can be used and incorporated by other work to help make energy-aware decisions when, for example, designing or optimising Cloud applications and efficiently managing the Cloud resources.

In order to achieve this aim, a number of objectives are identified, which mainly include:

- ***O.1:*** *Exploring the current energy efficiency related issues and challenges in the Cloud paradigm.* Improving the energy efficiency in Clouds has been an active research area. Therefore, it is important to understand the current challenges in order to contribute with a solution that can be used to help addressing these challenges.
- ***O.2:*** *Investigating how the energy usage of Cloud services can be identified in a Cloud environment.* The energy consumption can be easily identified at the PM level, but is not directly measured at the VM level. This work therefore introduces and implements an energy-aware Cloud system architecture that can profile the energy usage at both physical and virtual levels in a Cloud environment.
- ***O.3:*** *Exploring and understanding how the physical resources are correlated with the virtual resources usage and their impact on energy consumption*. This work characterises the physical and virtual resources usage with direct experimentation in order to identify the key parameters correlated with the energy consumption.

- ***O.4:*** *Investigating the use of mathematical modelling in this research context.* Hence, a new energy-aware profiling model is introduced to attribute PM’s energy consumption to VMs.
- ***O.5:*** *Investigating and exploring different workload patterns experienced by Cloud applications.* This is important in order to study and map the energy usage for each specific workload pattern.
- ***O.6:*** *Exploring machine learning techniques and prediction methods to forecast future workload and energy consumption.* This work introduces an energy-aware prediction framework to predict future energy consumption for VMs prior to deployment based on historical time-series workload patterns.

1.4 Research Methodology

This research has undergone a number of stages. The first stage is examining the issues of energy efficiency in Cloud Computing and the identification of a research opportunity, which is the need of energy-awareness at VM level. Then introducing an energy-aware system architecture as a solution to fulfil this need takes place, followed by the development of an energy-aware profiling model to attribute the PM’s energy consumption to VMs. The final stage is the introduction of an energy-ware prediction framework to enable energy prediction for both PMs and VMs in a Cloud environment.

In order to achieve the aim and objectives of this research, a scientific research method has to be used and followed. There are two main approaches that can be followed to do research, quantitative and qualitative [15]; the former is mostly used for research involving measuring variables and examining their relationship with statistical analysis, and the latter is mostly used for research involving exploring and understanding problems with descriptive analysis. This research has followed a quantitative approach with repeatable empirical experiments. To conduct research within the distributed systems domain, three methods are available:

- *Direct Experiments* [16], [17]: This method can be described as the use of a real environment, e.g. a testbed, for conducting experiments to validate a hypothesis or a solution. This method can give very accurate and reliable results but can be limited to the resources availability, time and effort to conduct such repeatable experiments. Hence, it can be costly and difficult to conduct large-scale experiments in a real environment [18].
- *Mathematical Modelling* [19], [20]: This method can be defined as the formulation of mathematical models that can describe a system and the relation and behaviours of different parameters within a system. Mathematical models usually consist of symbols and operations and can be used for different purposes, like training, estimation and prediction of behaviours within a system. The models developed in this method can be validated with experiments conducted in a real environment (direct experiment) or in a simulation [21].
- *Simulation* [18], [22]*:* This method can be defined as the use of a simulated environment for conducting experiments to validate a hypothesis or a solution. This method is based on imitating and emulating the real system, and it can offer performing experiments in a short time. Also, this method can enable performing large-scale experiments with low cost and effort. However, while the nature of this method involves some randomness, it gives less accuracy and reliability of the results as

compared to direct experiments on a real environment. Hence, another limitation of the simulation is that it needs further verification in terms of representing the real environment [23]. The simulation methods can be validated with mathematical models or direct implementation in a real environment [21].

For this research, both mathematical modelling and direct experiments methods are used. The energy-aware profiling and prediction models presented in this thesis are formulated using mathematical modelling. Direct experiments are also used and conducted on a Cloud testbed to verify and validate the applicability of these models on a real Cloud environment.

The simulation method has not been considered in this thesis for a twofold reason. Firstly, the experimental results obtained in a simulation can be less accurate as compared in a real environment. Secondly, it is difficult to learn the real behaviour and correlation of the Cloud resources using simulation. To illustrate, by conducting some direct experiments in this thesis using a real Cloud testbed has led to identifying the required parameters for the development of mathematical models, as to be presented in Sections 5.2.1.1 and 6.3.1. Though, the simulation method can be considered in future work to further examine the scalability-related issues, which is difficult to address in a testbed with limited resources.

1.5 Main Contributions

The main contributions of this thesis are the following:

- *An energy-aware Cloud system architecture.* This architecture includes the required components to address the first research question (**Q.1**) by enabling energy-awareness at the deployment and operational levels in a Cloud environment.

- *An energy-aware profiling model.* This model is developed with the use of mathematical modelling, and is aimed to address the second research question (**Q.2**) by enabling energy-awareness at the VM level by attributing the energy consumption to heterogeneous and homogeneous VMs running on the same PM based on the size and CPU utilisation of each VM.
- *An energy-aware prediction framework.* This framework consists of a number of mathematical models with the aim of addressing the thirds research question (**Q.3**) by forecasting the future energy usage of VMs prior to service deployment. This framework first predicts the VMs' workload by considering the type of these VMs and their historical workload patterns using Auto-Regressive Integrated Moving Average (ARIMA) model. The predicted VM workload is then correlated to the physical resources within this framework in order to get the predicted VM energy consumption.

1.6 Thesis Overview

The remaining chapters of this thesis are organised as follows:

- **Chapter 2** presents an overview of the concepts of Cloud Computing, Cloud system architecture, Cloud application workload patterns and the issues of energy efficiency in Cloud Computing.
- **Chapter 3** reviews the literature and technologies for enhancing the energy efficiency in Cloud Computing. It begins with a discussion on the

requirements engineering and followed by discussions on programming models, energy-aware resource management, and energy-aware profiling and prediction models, all of which drive towards energy efficient Cloud Computing.

- **Chapter 4** introduces an energy-aware Cloud system architecture with thorough details of its main components and their interactions. This is followed by some experiments on a Cloud testbed to provide an early evaluation of this architecture in terms of enabling energy-awareness in a Cloud environment.
- **Chapter 5** presents the mathematical development of an energy-aware profiling model for enabling energy-awareness at the VM level in a Cloud environment, followed by a number of experiments on the Cloud testbed to evaluate the capability of the presented model.
- **Chapter 6** introduces an energy-aware prediction framework which consists of a number of mathematical models in order to forecast the energy consumption of VMs prior to service deployment. This is followed by a demonstration of some experiments on the Cloud testbed to evaluate the capability of the introduced framework.
- **Chapter 7** provides an overall evaluation of the research presented in this thesis.
- **Chapter 8** summarises the work and contributions presented in this thesis and discusses some future work directions.

Chapter 2 Background

2.1 Overview

This chapter presents the essential background of this research. It starts by introducing the concept of Cloud Computing with a detailed description of its definition, system architecture, services types, deployment types and virtualisation technology, as presented in Section 2.2. The aspects of Cloud applications are then discussed by describing their properties, design patterns and workload patterns, as presented in Section 2.3. The energy consumption and energy efficiency issues in Cloud Computing are also presented in Section 2.4.1. This chapter then concludes with a discussion of some of the streams towards addressing these issues and enhancing the energy efficiency in Cloud Computing, as presented in Section 2.4.2.

2.2 Cloud Computing

Cloud Computing has changed the way businesses and individuals use IT resources. Today, instead of buying their own IT resources, they can use hardware and software as services offered by Cloud Computing providers with reasonable costs, based on pay-per-use model, and not worry about the overheads associated with the total cost of ownership. In 2010, Cloud Computing has been considered as a strategic shift point in IT after the last shift in 1994, which was the wide adoption of the Internet [24]. Additionally, it has been argued that with the increasingly perceived common vision, Cloud Computing has become the fifth utility after gas, water, electricity and telephony providing the basic level of computing services to be used by the general public in a daily basis [25]. Cloud Computing has evolved out from the extensive research on Grid

Computing. Grids are known as the backbone supporting the development of Clouds [26]. In terms of control and management, Clouds are centralised whereas Grids are decentralised [25], [27]. For usability, Clouds are considered to be user friendly while Grids are known to be difficult to use [27].

In the following subsections, Cloud Computing definition, system architecture, services types, deployment types, and virtualisation will be presented.

2.2.1 Definition

Cloud Computing is defined by NIST as:

> "*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable Computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction*" p.2, [28].

However, there is no standard definition for Cloud Computing. So, it has also been defined many times by different IT experts, each with different focus of Cloud aspects. Vaquero et al reviewed a number of different Cloud Computing definitions and offered a comprehensive definition which is a large pool of virtualised resources that can be easily used and accessed, re-configured and scaled dynamically, utilised, and based on a pay-per-use model and Service Level Agreements (SLAs), which is an electronic contract between end-users and service providers [27].

2.2.2 System Architecture

NIST has presented a general and high-level reference architecture model that considers all Cloud actors along with their roles and interactions in Cloud Computing. As shown in Figure 2-1, NIST Cloud Computing architecture model

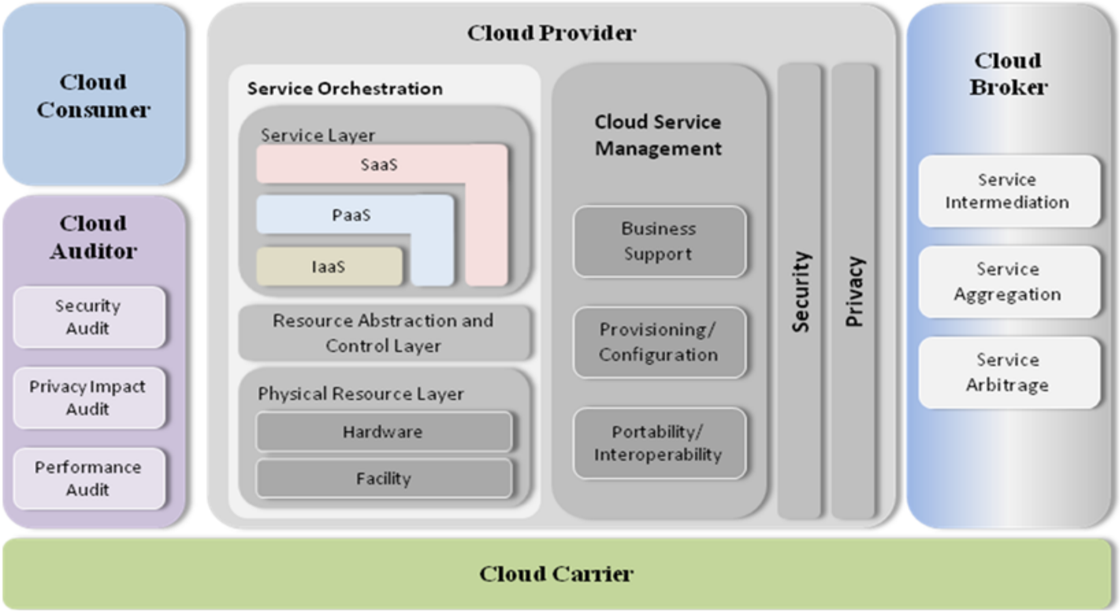

Figure 2-1: NIST Cloud Computing Reference Architecture Model [29]

mainly consists of five actors, namely Cloud consumer, auditor, provider, broker and carrier [29].

In terms of the roles and interactions, the Cloud consumer is an entity, an individual or an organisation, that can request and use any Cloud services from a Cloud provider, who is responsible for providing and maintaining the Cloud services, or through a Cloud broker, who has the role of negotiation between the provider and consumer and managing the delivery of Cloud services. The Cloud auditor can have the role of collecting essential information in order to assess the delivery and implementation of Cloud services. Finally, the Cloud carrier is responsible for providing the network connectivity in order to facilitate the communication between the actors and the transportation of Cloud services [29].

Moving on to the architectural layers of Cloud Computing, Buyya et al [30] stated that the Cloud architecture consists of mainly three principal layers, namely system level, core middleware, and user-level middleware, as shown in Figure 2-2.

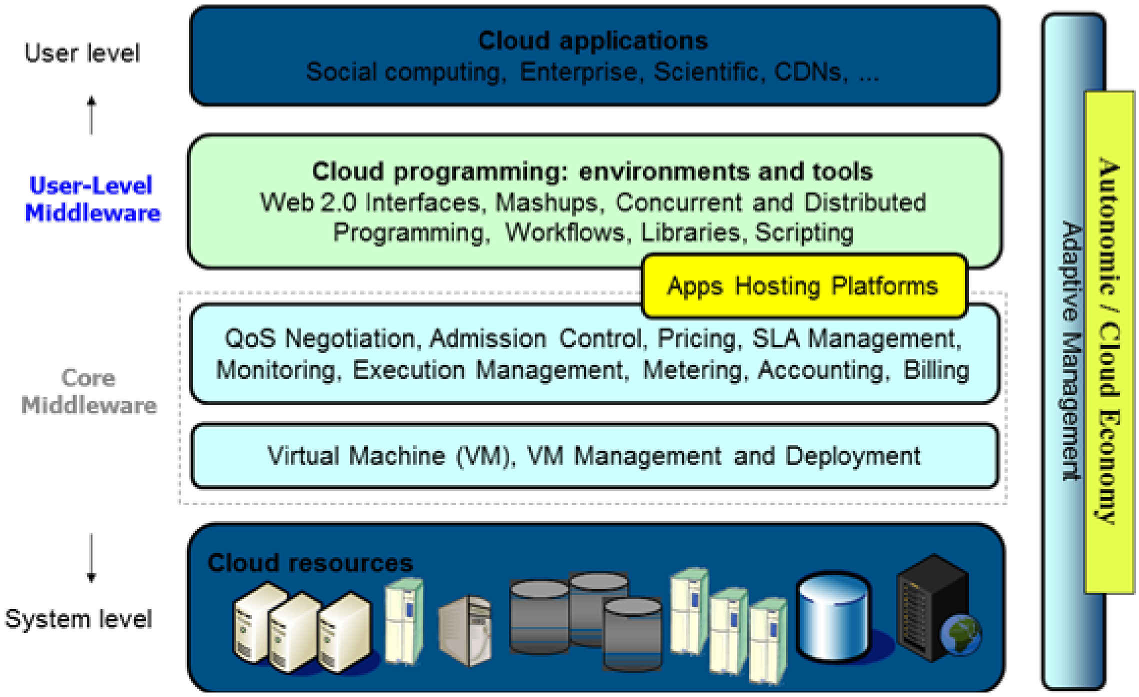
User level
User-Level Middleware
Core Middleware
System level
Cloud applications
Social computing, Enterprise, Scientific, CDNs, ...
Cloud programming: environments and tools
Web 2.0 Interfaces, Mashups, Concurrent and Distributed Programming, Workflows, Libraries, Scripting
Apps Hosting Platforms
QoS Negotiation, Admission Control, Pricing, SLA Management, Monitoring, Execution Management, Metering, Accounting, Billing
Virtual Machine (VM), VM Management and Deployment
Cloud resources
Autonomic / Cloud Economy
Adaptive Management

Figure 2-2: Layered Cloud Computing Architecture [30]

Starting from the lower level, the system level forms the basis of the Cloud architecture where all the physical resources, like servers, are set, and these resources are controlled by the virtualisation services that exist above this layer [30]. Further, the core-middleware layer is the platform that sets the run-time environment to host and control the application services at the user-level. Moreover, software frameworks exist at the user-level middleware to support the developers to create an environment for applications' development and execution in the Clouds [30]. Finally, the Cloud application layer contains the applications that can be deployed and accessed directly by the end-users [18].

Furthermore, Zhang et al [10] divided the Cloud Computing architecture into four main layers, namely hardware, infrastructure, platform and application layers, as depicted in Figure 2-3.

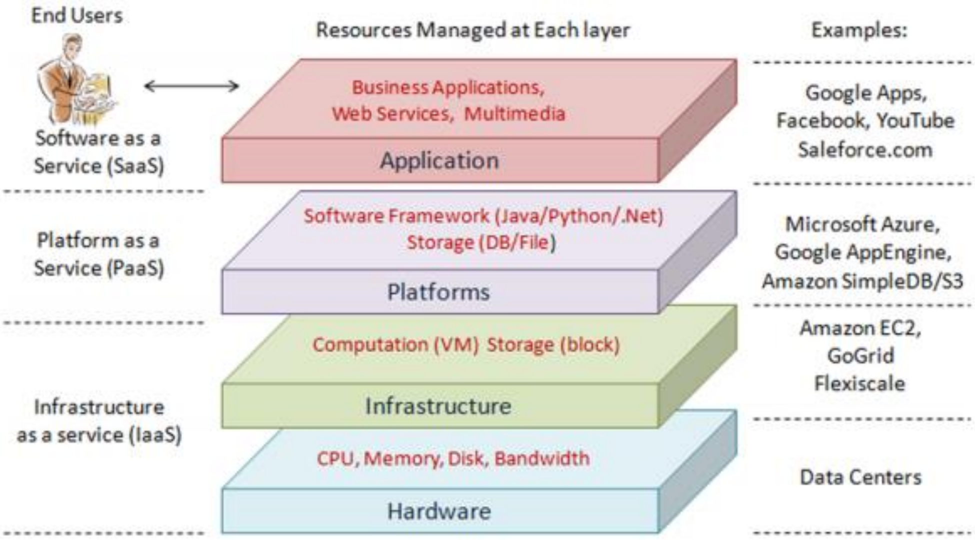
End Users
Resources Managed at Each layer
Examples:
Business Applications, Web Services, Multimedia
Application
Software as a Service (SaaS)
Google Apps, Facebook, YouTube Saleforce.com
Software Framework (Java/Python/.Net) Storage (DB/File)
Platforms
Platform as a Service (PaaS)
Microsoft Azure, Google AppEngine, Amazon SimpleDB/S3
Computation (VM) Storage (block)
Infrastructure
Infrastructure as a service (IaaS)
Amazon EC2, GoGrid Flexiscale
CPU, Memory, Disk, Bandwidth
Hardware
Data Centers

Figure 2-3: Cloud Computing Architecture [10]

At the bottom of this architecture is the hardware layer where the Cloud physical resources, like servers, switches, routers, and cooling systems, are managed within data centres [10]. On top of the hardware comes the infrastructure layer, also called the virtualisation layer, which consists of a pool of virtualised computing resources through the use of virtualisation technologies. The application frameworks and operating systems are included in the platform layer, which provides the environment to deploy the applications in VMs. The actual Cloud applications sit on the top of the architecture, at the application layer. The key distinction of the Cloud architecture as compared with other traditional environments like dedicated server farms is that the layers of Cloud Computing are loosely coupled from each other and can work separately [10]. In essence all these Cloud Computing layers can be provided separately by different Cloud providers.

2.2.3 Services Types

Based on the Cloud architectural layers, there are three main types of Cloud services, namely Software as a Service (SaaS) supplied at the application layer (also known as user level), Platform as a Service (PaaS) supplied at the platform layer (also known as middleware level), and Infrastructure as a Service (IaaS) supplied at the infrastructure and hardware layers that consist of virtual and physical hardware (also known as system level) [10], [18].

As stated by NIST, in the SaaS service model, the end-users are able to access and use the Cloud applications offered by the providers, but they have limited control to configure user-specific application settings. Also, they do not have control or access to the underlying Cloud resources that host and underpin these applications. In the PaaS service model, the end-users can deploy their own applications on the provider’s Cloud infrastructure with a full control of the applications and their settings and perhaps some settings for the run-time environment hosting these applications; but they do not have control or access to the underlying Cloud resources, like storage, servers, network or operating systems. In the IaaS service model, the end-users can have access to use and provision some of the Cloud infrastructure resources, like computing resources, networks and storage, on which they can deploy their own applications and operating systems; but they do not have the control of the maintenance of the underlying Cloud resources, which is the responsibility of the providers [28].

SaaS is about offering ready-to-use software applications, like Google Documents, to the end-users without worrying about the platform and hardware hosting these applications. PaaS is about offering services, like Google App Engine, that provide the run-time environment to host end-users’ applications. IaaS is about offering virtualised resources, like storage, as services to be used by the end-users [31]. All of these three types of Cloud services can be provided by one or different Cloud providers.

2.2.4 Deployment Types

Cloud Computing can be deployed in many models, which can be mainly public, private, hybrid, and community Clouds [28]. Public Clouds offer open access of IT resources and services to all end-users, including individuals and other organisations, through the Internet and at reasonable prices that allow the end-users to save the cost of having in-house built resources. Public Cloud infrastructures are fully controlled on the premises of the providers offering the services to the end-users, and the end-users do not have control of how and where these infrastructures are being managed and hosted.

On the other hand, access to the IT resources and services offered by private Clouds are restricted only to those who own the Clouds and their subsidiaries, which can enhance the security aspect. Another benefit of deploying a private Cloud is to allow an organisation to utilise its internal IT resources efficiently through the use of virtualisation technology [32].

Moreover, when organisations need to scale up their private Clouds, they can outsource more IT services at low cost from the public Clouds, for example to fulfil their non-critical business needs from the public Clouds and keep their sensitive business data stored locally in their private Cloud; this type of combination is called Hybrid Clouds [24]. The formation of hybrid Clouds can be through combining two or more public, private, or community Cloud infrastructures [28].

Finally, a community Cloud is usually formed between a number of organisations with common interests to collaborate and share their Cloud infrastructures in order to achieve their missions. The Cloud infrastructures in this community deployment model can be owned and controlled by any of the organisations within the community, a third party, or a combination of them [28].

2.2.5 Virtualisation

Virtualisation has been defined as:

> "*a technology that combines or divides computing resources to present one or many operating environments using methodologies like hardware and software partitioning or aggregation, partial or complete machine simulation, emulation, time-sharing, and many others*" p.2 [33].

Hence, the main role of virtualisation is to abstract the physical hardware machines and provide virtualised machines that can work in isolation and run different applications and even operating systems. Therefore, virtualisation technology adds an important value to the Cloud infrastructure by allowing a better resource utilisation, cost reduction [34] and achieving significant energy savings [35].

As mentioned earlier when discussing the architecture of Clouds, the virtualisation layer is just set above the physical layer in order to make an abstraction between the hardware and software. So, virtualisation is considered as a crucial technology of Cloud Computing offering two important features, namely abstraction and encapsulation [26]. Virtualisation technology is used to enable sharing of physical resources to multiple virtual resources with abstracting the complexity and details of the physical resources and make them as a unified pool of resources [26], [34]. Via virtualisation technology, each application can be encapsulated to provide enhanced manageability, isolation and security when performing such operations like configuring, deploying, starting, suspending, migrating, resuming and stopping these applications [26].

A number of benefits can be achieved through the use of virtualisation technology in Cloud environments. Some of these benefits are allowing better resource utilisation, managing the servers easily, server consolidation and live VMs migration [36]. Therefore, the use of virtualisation in Cloud data centres can reduce the number of the needed physical resources, which would then lower the capital cost as well as reducing the power consumption and cooling systems [37]. To illustrate, many VMs can be created and run on a single physical machine via server consolidation. Live migration of the VMs to the underutilised physical machines would enable to turn-off more physical machines and allow maximum utilisation of the running physical machines, which would enhance the energy efficiency of the data centre [26], [36].

2.2.5.1 Virtual Infrastructure Manager

In order to build and deploy Cloud infrastructures, a Virtual Infrastructure Manager (VIM), also known as a Cloud Operating System (Cloud OS), is used by the Cloud infrastructure providers to manage the virtualised and physical resources and enable the provisioning of the virtualised resources based on the end-users requirements of the services. Some examples of the VIMs include OpenNebula [38], OpenStack [39], and CloudStack [40], all of which are discussed next.

OpenNebula is an open source toolkit that provides a platform for deploying private, public and hybrid Cloud infrastructures. Its architecture is organised in three main layers, namely drivers, core, and tools layers [41]. Starting from the bottom of the architecture, the drivers layer consists of physical infrastructure drivers that abstract the underlying physical resources. Also this layer includes other Cloud drivers to facilitate accessibility to remote Cloud providers. On top of this layer is the core layer which consists of a number of components, such as image manager, VM manager, information manager, storage manager and network manager. These core components rely on the drivers at the bottom layer in order to support the deployment, management and monitoring of the virtualised Cloud resources. Finally, the high-level tools layer is at the top of the architecture and consists of a number of components that facilitate different functionalities. Some of the high-level functionalities supported at this layer include initial placement of VMs on particular physical servers and access via Graphical User Interface (GUI) or Command-Line Interface (CLI) for both administrators and users to make different operations. Other functionalities supported at this layer also include the management of multi-tier services and admission control to reject or accept such a service and the use of different Application Programming interfaces (APIs) interfaces, like Open Virtualised Format (OVF), to enable interoperability and portability of OpenNebula and allow accessibility of its functionality to external consumers [41]. OVF [42] is an open standard format that facilitates packaging, distributing and defining virtual appliances and has been widely used in Cloud Computing as it can enable interoperability between different providers.

OpenStack is another open source VIM that is based on a modular architecture consisting of many interrelated components, developed separately in different projects, working together to deliver a platform for a complete deployment and management of public and private Cloud infrastructures [39]. Some of its main components include Compute (Nova) for handling the VM instances lifecycle [43]; Networking (Neutron) for enabling network connectivity between OpenStack components and providing APIs access to its users [44]; Object Storage (Swift) for providing highly fault-tolerant, redundant, and scalable object storage system [45]; Identity (Keystone) for providing authentication and authorisation services [46]; Image Service (Glance) for storing and retrieving disk images of VMs [47]; and Dashboard (Horizon) for providing a web-based interface for the users to access and interact with the underlying services of OpenStack and perform operations like instantiating a VM instance [48].

CloudStack is another open source platform that can be used as a VIM to manage and orchestrate a pool of computing, networking and storage resources and to deploy private, public or hybrid Cloud infrastructures [49]. Its architectural deployment mainly comprises of two main components, Management Server and the Cloud Infrastructure [50]. The Management Server acts as the main controller of the Cloud deployment and has a number of functionalities including allocating VM instances to hosts, assigning public and private Internet Protocol (IP) addresses and storage to VM instances, providing access for both end users and administrators through web interface and APIs, and managing templates, ISO images and snapshots. The Cloud infrastructure consists of all other resources, such as storage devices, hypervisors, IP address blocks and VLANs, to be managed by the Management Server. CloudStack can deploy a set of management servers to control scalable and largely distributed Cloud infrastructures with the use of any networking technologies, like VLANs and VPNs [50].

OpenNebula, OpenStack and CloudStack have a common role of providing a platform for deploying compute, storage and networking resources and allowing management and provisioning of these resources via a Web interface, command-line or APIs. However, there are differences in terms of their capabilities and performance based on the configurations, settings and size of their deployment. For example, OpenStack has many components and installing all of them would introduce an overhead to manage them and may render the performance [51]. Therefore, in order to get the best performance when using OpenStack, the administrator should only install the required components to fulfil the needs of their Cloud deployment. In comparison, OpenNebula does not have such limitations as it provides a centralised deployment and has fine-grained core [51]. There are other VIMs available freely or commercially for the deployment and management of Cloud infrastructures, like Eucalyptus [52], Nimbus [53], oVirt [54], VMware vSphere [55] and many others.

2.2.5.2 Hypervisors

The Virtual Machine Monitor or Manager (VMM), known as the hypervisor, is the main component in a Cloud environment that is responsible for managing and controlling the VMs' operations including creating, running, migrating, copying, and deleting the VMs [34]. Hypervisor-based virtualisation abstracts the underlying resources to provide virtualised instances known as VMs which can have and run their own and complete OS [56], [57]. Hypervisors can be implemented in different levels, like full virtualisation and hardware virtualisation. Full virtualisation is implemented when the hypervisor runs on top of the underlying host OS, and hardware virtualisation is implemented when the hypervisor runs directly on top of the underlying physical hardware [51]. Some examples of hypervisors include KVM [58], [59], Xen [60] and VMware [61].

2.2.5.3 Containers

Another type of virtualisation can be based on containers technology, which is considered a lightweight substitute in comparison to hypervisors [56]. Container-based virtualisation, also called the OS-level virtualisation, modifies the underlying host OS to provide isolated instances, called containers, that can run different applications all together by sharing the host OS [57], [62]. Containers can also run on the VMs OS providing further virtualised isolated instances at the PaaS layer. When containers run inside the physical host OS, the overhead resulting from managing the virtualisation layer of the VMs created by the hypervisors is reduced [51]. In terms of performance, containers-based virtualisation is better than hypervisor-based virtualisation as there is a small overhead for the hypervisor to translate the instructions of the guest OS at the VMs to the host OS, while the containers running directly on the host OS can achieve almost native performance of host OS [57]. In terms of isolation, hypervisor-based is better than container-based virtualisation because each VM can run in an isolated guest OS, while containers share the host OS [57]. Hence, containers technology also restricts the flexibility of supporting and running different applications requiring different OS, while in hypervisor technology each application can run on different VMs with its own guest OS. Some examples of containers include Linux Containers (LXC) [63], Docker [64] and Warden Container [65].

2.3 Cloud Computing Applications

2.3.1 Properties

The properties of Cloud applications are derived from the characteristics of Cloud Computing. As stated by Fehling et al [66], Cloud Computing applications should be designed differently from the traditional software applications; in the essence that they should exploit the properties of Cloud Computing. So, Cloud applications should be able to support the characteristics of Isolate state, Distribution, Elasticity, Automated management, and Loose coupling (IDEAL), all of which reflect the patterns of Cloud Computing environments.

Firstly, Cloud applications should allow distribution by nature; so the Cloud-native applications should be separated into application components to support distribution among Cloud resources [67]. Also, the Cloud-native applications should support elasticity to allow dynamic reservation and release of the Cloud resources to alter the performance rapidly based on the changes of the workloads. So, to adjust the performance according to the increase of workloads, these applications should support scaling out (horizontal scaling) by increasing the number of the assigned Cloud resources and support scaling up (vertical scaling) by increasing the capability of the assigned Cloud resources that run the applications. In terms of isolated state, large portions of Cloud application components should be designed to be stateless in order to automatically support scaling the application more easily with less management for handling the resources state when adding or removing any resources. Additionally, automated management should be supported by the Cloud-native applications to flourish the elasticity of continuously adding and removing resources as needed. Finally, the Cloud-native application should support loose coupling in the essence that the dependencies should be minimized between its components since they run on a number of resources that may change constantly [66].

2.3.2 Design Patterns

In the field of Computer Science, a pattern refers to the abstraction of a solution to commonly reoccurring problems in different contexts [68]. Design patterns refer to abstraction of structured and reusable software design solutions to solve common problems with description of how they can be applied [69]. Similarly, Cloud patterns, as a further pattern category, refer to the abstraction and description of decent solutions to repeatedly common problems in relation to Cloud Computing [70]–[72]. These patterns are needed to help alleviate the software design challenges [68], for example when migrating traditional applications towards Cloud-native applications that should exploit the nature and properties of Cloud Computing.

The application and implementation of these design patterns can be through the use of one or many mechanisms. A mechanism is considered as a ready-to-use technology artefact, like a hypervisor. When a number of design patterns are combined, they form a compound design pattern that can deliver more granular solution, like IaaS [73].

2.3.3 Workload Patterns

There are a large number of different Cloud applications with different requirements of resources. Depending on the behaviour of users and submitted tasks, the Cloud applications can experience different patterns of workloads, which are depicted based on the utilisation of the IT resources hosting the applications. These workloads can be categorised as static workload, periodic workload, once-in-a-lifetime workload, unpredictable workload, and continuously changing workload, as discussed in [66], [74].

As shown in Figure 2-4, a static workload pattern occurs when an application is running continuously with the same workload resulting in equal utilisation of Cloud resources over a period of time. A periodic workload pattern can be experienced when an application is running with a peak interval that repeatedly happens over time.

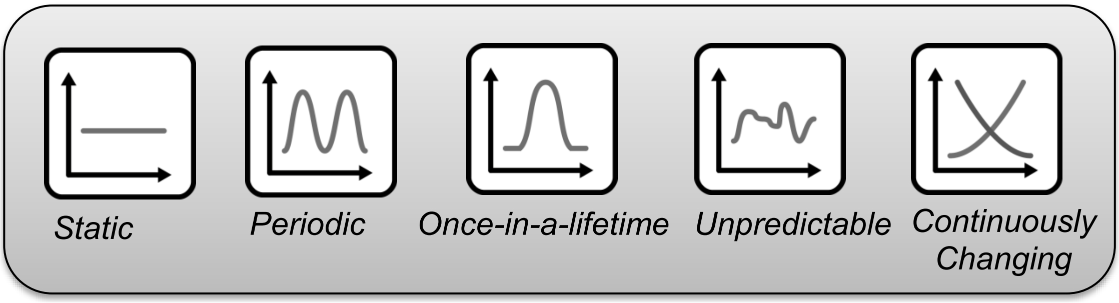

Figure 2-4: Cloud Application Workload Patterns [74]

Further, when an application is running with equal utilisation of resources and has only one peak utilisation over time, it is considered a once-in-a-lifetime workload pattern. An unpredicted workload pattern arises when an application has a random peak utilisation over time. Finally, when the application has a constant increase or decrease of resources utilisation over time, it experiences a continuously changing workload pattern [66], [75]. These different types of application workload patterns can have a different impact of energy consumption depending on the usage of the physical resources. The static and periodic workload patterns of Cloud applications are considered in this thesis.

2.4 Energy Efficiency in Cloud Computing

2.4.1 Energy Consumptions in Clouds

The scalability of Cloud Computing is considered one of its main advantages supporting dynamic increase and decrease of the computing resources to meet the end-users demands. Cloud Computing data centres are commonly known as large-scale environments equipped with thousands of servers that consume considerably large amounts of energy in order to operate. Also, the cooling systems operating within the data centres consume a substantial amount of energy as well.

Thus, with the wide adoption of Cloud Computing, energy consumption cost has become one of the main issues for Cloud providers to maintain. For economic aspects, a data centre may consume about 100 times more energy compared to a typical office with the same size [2]. In comparison with the electricity demand of global countries in 2011, Cloud Computing was ranked the sixth largest electricity consumer with 684 billion Kilowatt-Hour (kWh), just after China, US, Japan, India, and Russia [76]. As the online population and the use of the Internet increase gradually, this electricity demand of Cloud Computing is estimated to increase by 60% or even more by 2020 [76]. Therefore, the energy consumption has become one of the greatest cost factors for Cloud computing vendors who consider energy efficiency as a vital issue [3].

Further, the increase of energy consumption and CO2 emissions of Cloud infrastructures has become a vital concern in relation to the environmental sustainability [77]. So, Cloud vendors face huge pressure from governments and other organisations to reduce the CO2 emission from their data centres to have less impact on the environment. Thus, energy consumption in Cloud Computing infrastructures is considered a significant concern in terms of both economic and ecological perspectives, which leads to different streams being emerged towards enhancing the energy efficiency in Cloud environments.

2.4.2 Streams of Enhancing Energy Efficiency in Clouds

Some studies have investigated different ways for improving the energy efficiency of Cloud Computing throughout various streams, such as, energy-aware resource deployment and management, programming models, requirements engineering, and energy awareness modelling by introducing new energy-aware profiling and prediction models. For instance, in energy-aware resource management, the focus is on introducing and using different techniques, like powering-off idle servers, VM consolidation, and energy-aware scheduling, to efficiently manage the Cloud resources with less energy usage. In terms of requirements engineering and programming models, the focus is on how to provide suitable tools and environments in order to design energy efficient software that would consume less energy when running on the underlying physical resources. Thus, different energy efficient techniques have been introduced in different streams to help the Cloud providers reduce the energy consumption cost of their infrastructure, which can then lead to reducing the cost of operational expenditure and having less impact on the environment.

More details and review of those different streams of enhancing the energy efficiency in Cloud Computing are discussed in the following Chapter 3.

2.5 Summary

This chapter has introduced some essential aspects and background about Cloud Computing including its definition, system architecture, services types, deployment types and virtualisation technology. The properties, design patterns and workload patterns of Cloud applications have been also presented. Finally, this chapter has concluded by discussing the issues of energy consumption and energy efficiency in Clouds, and how the current research streams have been driven towards addressing these issues.

Chapter 3 Energy Efficiency in Cloud Computing

3.1 Overview

This chapter reviews the literature towards energy efficient Cloud Computing. It first discusses the related work on energy-aware computing, including requirements engineering, programming models, energy-aware resource management, energy efficiency metrics and finally discusses economic aspect on energy-aware pricing, as presented in Section 3.2. Then, it reviews the related work on energy aware profiling and modelling for PMs and VMs in Cloud environments, along with forecasting models for future prediction of the workload and energy usage, as presented in Section 3.3. Finally, a summarised discussion of the closely related work is presented in Section 3.3.5.

3.2 Energy-Aware Computing

3.2.1 Overview

As Cloud data centres consist of large computing resources consuming a large amount of energy in order to operate, enhancing the energy efficiency has gained a significant interest in both academia and industry because of the high associated impact of excessive energy usage on economic, environment, and performance [78]. For example, the cost of energy usage to run a data centre is estimated to double in every five years [79]. Hence, the cost of energy has been considered as one of the greatest expenses contributing to increased cost of ownership for data centres [16], [80]. Also, the excessive usage of energy in data centres causes environmental issues [81], [82]. For instance, gas emission caused by the ICT industry is predicted to be accountable for 2.3% of the global emission in 2020 [83]. To address this ICT gas emission issue, more investment is expected to be put towards using and adapting energy efficient systems in ICT, which predictably could drive this footprint percentage to decrease to 1.97% by 2030 [84]. In terms of performance, it can be difficult to achieve high performance and energy savings at the same time [78]. But, most of the focus in this regard is about finding a balanced trade-off between energy and performance, which is still challenging to achieve as different end-users may have different preference between having high performance or saving more energy costs and sacrificing some of the performance.

Thus, a number of streams have been investigated at different layers of Cloud Computing to ensure efficient operations and management with energy awareness in mind. For instance, in requirements engineering, some work emphasised the importance of incorporating energy information and specifying energy goals within the requirements and ensuring the applications adapt in accordance with these goals. Also, some work introduced energy-aware programming models so that developers can make use of and write energy efficient code that would consume less energy when operating. Additionally, other streams presented different techniques to efficiently deploy the Cloud services and manage resources with consideration of energy efficiency. Furthermore, a number of metrics have been introduced to identify the energy usage and evaluate the energy efficiency of Clouds at different layers. For economic perspective, some work introduced new pricing mechanisms of the Cloud services with consideration of the energy usage. All of these different streams will be discussed in the following subsections.

3.2.2 Requirement Engineering

Software systems have advanced to self-adaptive systems to meet the growing needs for autonomic computing, which is about automatic management and adaptation to overcome unpredictable changes within computing systems. Self-adaptive systems are capable to adapt themselves by configuring and reconfiguring, augmenting their functionality, optimising, protecting, and recovering without the users' interactions [85]. These self-adaptive systems have been mostly developed through addressing design-time solutions to provide adaptation at run-time, while requirements engineering for self-adaptive systems has gained less consideration [86]. Therefore, a number of approaches have been introduced to model goal-oriented requirements engineering to support self-adaptation at run-time [87]–[89].

Nonetheless, requirements engineering that considers energy aspects has received less attention. Thus, there is a need to support the requirements engineering and design modelling to develop self-adaptive systems that certify energy-awareness at different layers of Clouds. Ponsard et al. [11] emphasised the need to consider energy efficiency at the application layer of Cloud Computing, and introduced a UML-based framework that can relate energy goals at the requirements level to be captured at the design level and also monitored at the run-time level. The aim of their framework is to provide the Cloud application developers with energy-awareness information at the design time to select the best trade-off between energy and overall performance. The application can then adapt based on the selected trade-off during the run-time. Thus, Cloud application analysts can use goal-oriented approach to specify energy goals within the requirements to be followed when designing, developing and deploying the applications. Though, energy information feedback from the service operational level is still needed to help set these energy goals.

3.2.3 Programming Models

In order to address energy efficiency from early stages at the application development level, the developers should use programming models that provide energy awareness and efficiency information of the underlying infrastructures when constructing the applications. With the increase mismatch between the programming models and the underlying hardware architecture, Shalf [90] highlighted the need to have programming models that reflect the underlying hardware. Also, the cause of this mismatch, as stated by Shalf [90], is power constrained nature of future hardware architectures. So, the programming models should be designed in a way to be efficiently compatible with the underlying architecture.

Xian et al [12] presented a general-purpose programing environment to simplify and help the developers make energy-efficient decisions for constructing energy-aware applications. Their framework requires in-depth knowledge about the logic of the application and offers different plans to get the desired functionalities for the execution of applications in accordance with their power costs. This programming framework offers an interface that attains the estimated energy consumption for selecting a specific plan [12]. Nevertheless, Oriaku and Lami in [91] argued that this framework may not be suitable for Cloud Computing services as it can increase the environmental cost of a late application deployment.

Schubert et al [92] state that the developers lack the tools that indicate where the energy-hungry sections are located in their code and help them optimize their code for enhancing energy consumption accurately instead of just relying on their own intuitions. So, in their work, they proposed *eprof*, which is a software profiler that narrates energy consumption to code locations; therefore, it would also help developers make better energy-aware decisions when they re-write their code [92]. For example, with storing data on a disk, software developers might choose between storing the data in an uncompressed format or a compressed format, which would require more CPU resources. Compressed data has been commonly suggested as a way to reduce the amount of I/O needed to be performed and therefore reducing the energy based on the hypothesis that the CPU can process the task of compression and decompression with less energy than the task of transferring large data from and to the disk [93]. However, that would depend on the data being processed. In fact, some conducted experiments in [92] with *eprof* profiling tool show that the process of compressing and decompressing the data significantly consume more energy than the process of transferring large amount of uncompressed data because the former would use more CPU resources than the latter. So, it can be a controversial issue depending on the application domains. Thus, having such tools identifying how the energy has been consumed would help the software developers to make energy-aware decisions.

Moreover, a framework called *Symbolic Execution and Energy Profiles (SEEP)* has been introduced in [13] as an approach to help the software developers make well informed decisions for energy optimisation from early stages at the code level. *SEEP* is designed to provide the developers with energy estimations to make them energy-aware while they are programming. So, instead of analysing the program code after it has been developed and deployed to identify the hot spots for high energy consumption, this framework aims to simplify the process of energy-aware programming from early stages during software development [13].

For Cloud Computing, there are a number of frameworks that are used for the development and deployment of Cloud applications and services. Some of these are Hadoop [94], Windows Azure [95], Microsoft Daytona [96], Twister [97], Manjrasoft Aneka [98], and Google App Engine [99]. However, energy efficiency is not considered in these frameworks. Thus, there is a need to have such a framework that would enable programming of applications and services in Clouds and take into account energy-aware requirements and software design appropriate for Cloud architecture.

3.2.4 Energy-Aware Resource Management

With the increase of Cloud applications and users, managing the Cloud data centre has become challenging for the operators to improve its energy efficiency without performance degradation. In terms of energy efficiency, it can be said that data centre A is more efficient than data centre B if A can process the same workload as B but with less power consumption, or A can process more workload than B with the same power consumption [100]. The challenge is even increased to address the issue of excessive energy consumption by a server may result in higher temperature, which can compromise its reliability and availability [14]. Also, the energy consumption of an idle server is considered wasted energy as it is used without any beneficial output [101]. Therefore, some consolidation techniques are employed to efficiently manage the Cloud resources and turn off unused idle servers. Yet, from the Quality of Service (QoS) perspective, turning off the idle servers can be considered risky in a dynamic environment because it may affect their availability as they would need some time to be turned on again and meet the new demand [14].

Previous research has attempted to tackle this challenge by introducing policies and techniques that can dynamically adapt to reduce the energy usage and enhance the energy efficiency of the data centre. For example, Data Voltage and Frequency Scaling (DVFS) technique alters the CPU power supply of voltage and frequency in accordance with the offered workload, which would then enable controlling one-third of the energy consumed by servers as it depends on the CPU utilisation [102]. Also, deploying Dynamic Power Management (DPM) can even save more energy by powering down all servers' components including CPU, memory, and disks. However, it would also increase the overhead to power these servers back on [102]. Additionally, Chawarut and Woraphon [103] proposed a CPU re-allocation algorithm, that combines both DVFS and live migration techniques, to reduce the energy consumption and increase the performance of applications in Cloud data centres. As shown on their simulation results, they argued that their proposed algorithm can decrease the energy consumption and execution time of the running Cloud services.

Moreover, Beloglazov et al proposed VM consolidation policies to optimise the resources utilisation of the hosts by migrating VMs from one host to another host [14]. Basically, in order to identify from which host the VMs should be migrated, upper and lower CPU utilisation thresholds for each host are set. When the CPU utilisation of a host exceeds the upper threshold, some VMs should be migrated to another host to prevent SLAs violations. On the other side, when the CPU utilisation goes below the lower threshold, all the VMs on that host should be migrated to another host in order to switch that host to sleep mode and save some energy from idle power consumption. Furthermore, in order to identify which VMs should be selected for migration, some VM selection policies have also been proposed [14]. When the upper CPU utilisation threshold of a host is violated, a Minimization of Migration (MM) policy selects the minimum number of VMs to migrate to another host in order to keep the utilisation below the upper threshold. Also, if the upper threshold is violated, a Highest Potential Growth (HPG) policy selects the VMs having the lowest usage of CPU relative to the capacity of CPU to reduce the possible increase of host's utilisation and avoid SLA violations. Furthermore, a Random Choice (RC) policy depends on a random selection of VMs needed to reduce the CPU utilisation of a host under the upper threshold [14]. However, their work does not consider the energy consumption overhead of VM consolidation.

Further, some studies have been conducted to reduce the energy consumption via resource management of the Cloud infrastructure without affecting the performance of the running services. For example, Lee and Zomaya [104] proposed two energy-conscious task consolidation heuristics to save energy by maximising resource utilisation and taking into account idle and active energy consumption. The proposed heuristics aim to assign tasks to the resources with minimized energy consumption and without performance degradation. They argue that the results of their experiments show significant energy saving. However, they simply assume in their energy model that there is only a linear increasing relationship between the PM CPU utilisation and energy consumption.

Moreover, Jung et al [20] introduced a holistic framework called *Mistral* that optimises the power consumption, application performance benefits and the overhead costs acquired by the dynamic adaptation and actions of the framework. Their approach focuses on improving the power consumption of the physical host, but it does not take into account the effect of particular workloads running on particular hardware with different performance characteristics.

Tchernykh et al. [105] presented an experimental study for several online job scheduling strategies in a Cloud environment with different workloads. In the experimental results, they used and analysed eight allocation strategies based on three group categories, namely, knowledge-free, energy-aware, and speed-aware. Knowledge-free scheduling strategy requires no information from the application (submitted jobs by users) or from the underlying resources; energy-aware strategy requires information about the energy efficiency and power consumption of the underlying machines; speed-aware strategy requires information about the speed and performance of the underlying machine. Considering the two metrics provider income and power consumption, the results reveal that the strategy of allocating jobs to the machine with the least power consumption (Min-e) outperforms the other allocation strategies [105]. The energy model used in this work simply considers summing up the machine's idle power and the extra variable power, which depends on the workload. However, the workload is not considered in this energy model when calculating the variable power consumption. Also, the workload used in their work is based on HPC jobs based on parallel and Grid environments and not on real Cloud environments as elasticity and virtualization aspects are not considered.

3.2.5 Energy Efficiency Metrics

Energy efficiency in Clouds can be assessed by different metrics. Most of the proposed metrics nowadays focus on assessing the energy efficiency in physical Cloud infrastructures while there is also a need for assessing the energy efficiency of Clouds in the other layers, like virtualisation and application layers. Therefore, the Cloud application software analysts and developers can make use of these metrics at the application layer to enhance their decision-making, for example, when setting the requirements of, designing, and developing the applications with consideration of energy efficiency. Also, the applications can be designed and developed to adapt during run-time in accordance with the energy efficiency targets. Further, the Cloud service providers can make use of these metrics in a twofold purpose. First, it can help them as an input to their techniques and strategies enhance their decisions to make energy efficient deployment of the Cloud services on the Cloud infrastructure resources as well as to make energy efficient management of these resources during the service operation. Second, it can also help them to evaluate the output of their deployment and resource management techniques with regards to energy efficiency.

In terms of Cloud infrastructure, there are some high-level metrics used to measure the energy efficiency in data centres. In addition to the well-known metric, Power Usage Effectiveness (PUE), Green Grid organisation has introduced other three metrics to help the Cloud data centre vendors and operators to improve the energy efficiency of their facilities [106]. Firstly, the Green Energy Coefficient (GEC) metric is used to calculate the amount of the facility's energy that comes from green sources. It is calculated as the green energy used by the data centre divided by the total energy consumption of the data centre. Secondly, the Energy Reuse Factor (ERF) metric is introduced to quantify the total energy that is exported from the data centre and reused somewhere else outside; and it is calculated as total reused energy divided by the total energy consumption of the data centre. Thirdly, the Carbon Usage

Effectiveness (CUE) metric provides an assessment of the total greenhouse gas emission from the data centre in relation to the total IT energy consumption of the data centre. It is computed as the total gas emissions divided by the total IT energy consumption [106].

However, despite the fact that the PUE metric has been successful and widely used, it has been argued that it is limited and used as an indicator for energy efficiency to the infrastructure management only and not considering the actual utilisation and optimisation of the computational resources to enhance the efficiency of the whole stack [107], [108]. Also, Bozzelli et al [109] have reviewed a number of software metrics and emphasised the importance to assess the energy efficiency not only form the hardware side but also from early stages of the software lifecycle in order to make such energy savings. Additionally, as stated by Wilke et al [110], analysing software's energy consumption is considered an important requirement for such optimisations. Therefore, Grosskop proposed a new metric called the Consumption Near Sweet-Spot (CNS) that identifies how well the system's energy efficiency optimum and its utilisation are allied by calculating the ratio between the average consumption and optimum consumption for a system to deliver a particular unit of work [107].

Moreover, other works have looked at other metrics for energy efficiency measurements, like utilisation percentage and SLA violation percentage. For example, in the work conducted by Beloglazov et al [14], they evaluated the efficiency and performance of their proposed resource scheduling algorithms by using some metrics, namely the total energy consumption, the average SLA violation, and the number of VM migrations.

Some work introduced models to measure the energy consumption in more details, like measuring energy consumption for each VM inferred from energy consumption of PMs in which they are hosted. These models have been developed based on different mechanisms, like on performance event counters, lookup table, or utilisation of resources, all of which will be presented in Section 3.3.3.

All in all, as mentioned earlier, the use of different metrics to consider and assess the energy usage and energy efficiency of Clouds from different layers other than the physical infrastructures only can be beneficial for different stakeholders, including the Cloud application software analysts and developers as well as the Cloud service providers.

3.2.6 Energy-Aware Pricing

In terms of economic aspect, Cloud Computing has been considered as a business model that offers services to the users based on what they use [27]. By using Cloud services, the users can save the cost of Capital Expenditure (CAEX) for buying their own IT resources and the cost of Operational Expenditure (OPEX) for maintaining these resources. Therefore, the costs of CAEX and OPEX in the Cloud delivery model reside on the service providers.

The energy consumption of Cloud infrastructures resources is considered one of the greatest cost factors to maintain by the service providers [3]. Current pricing models used by the service providers are based only on the usage of the virtualized resources, like CPU, memory, and disk, and do not consider the energy consumed by these resources. For instance, Microsoft Azure Virtual Machines [111] and Amazon Elastic Compute Cloud (Amazon EC2) [112] charge the consumers for the offered services on a timely basis based on the resources’ usage, but without consideration of the energy consumption. In order to properly alleviate the cost of OPEX for Cloud service providers and offer transparent pricing, energy usage should be considered when designing pricing mechanisms for the offered services based on how and when these services are used. For example, if these services were extensively used only during the peak times, the operation would cost more because the electricity cost would go higher during these times. Thus, energy-aware considerations should be taken into account when designing pricing mechanisms for Cloud services in order to efficiently contribute to the overall business model of Cloud Computing. In order to introduce energy-aware pricing models for Cloud services, the energy information has to be identified not only at the physical level but also at the virtual level as different VMs can be owned by different customers and run on the same PM.

There are many research conducted to model pricing mechanisms for the offered Cloud services [113]. However, their approach is still limited in the essence that it does not consider the actual cost of energy. With the increasing electricity cost of the data centre to the point that it can often surpasses the cost of IT equipment over a period of time [114], the power consumption has become a vital concern for Cloud service providers. Thus, a number of work has introduced new pricing mechanism for the offered services to be aligned with the actual energy costs [3], [115]–[117]. For instance, an approach has been presented by Mukherjee et al [3] to model the users' behaviour by making them energy-aware when they make decisions for their service configurations. To illustrate, they introduce an economic model that allows a user to choose an acceptable configuration for the service based on a green point rating. Then, discounting is performed dynamically based on the green point. The greener the service configuration is selected the higher discount the customer is offered. So, prices of the services can be offered differently based on the customers' class, which would enable the Cloud vendors to maximise their profits while providing more discounts [3].

Moreover, another study presented by Narayan and Rao [115] proposed a pricing mechanism that maps between the cost of electricity input to the infrastructure and the output cost of the Cloud services. They claim that their pricing scheme fluctuates dynamically in accordance with the variation of the electrical input costs that are measured by a smart grid. Nonetheless, since the services are priced dynamically, it would elevate an issue with the price uncertainty that the end-users have to pay. So, they suggested that a pricing prediction model could be further integrated to overcome this limitation.

3.2.7 Overall Discussion

Energy consumption has become an important factor in Cloud Computing environments. Work presented in Sections 3.2.2 and 3.2.3 shows the importance of considering the energy usage information from early stages when specifying the requirements, designing and programming Cloud applications to make them ideally operate in an energy efficient way. In terms of the work on resources management presented in Section 3.2.4, energy consumption is considered either as an outcome to evaluate such resource management techniques or as an input to feed such decisions to efficiently manage the resources during the service operation time. In addition, various metrics that used energy usage as a main constituent part for evaluating the energy efficiency in Clouds have been reviewed in Section 3.2.5 concluding with the importance to have metrics to identify the energy usage not only for the PMs but also for VMs. Further, work discussed in Section 3.2.6 emphasises the significance of considering energy consumption in pricing mechanisms in order to make the consumers aware about their energy usage and increase the transparency of their charges. Also, as the energy consumption is considered a significant factor of the OPEX cost of maintaining Cloud infrastructures, energy-aware pricing mechanisms can help the providers alleviate the cost of energy.

Therefore, modelling and profiling energy consumption is needed in order to provide the awareness of the energy use in a Cloud environment. Some work has investigated in energy awareness and modelling, as discussed in the following Section 3.3.

3.3 Energy Awareness and Modelling

3.3.1 Overview

To begin with, the energy efficiency of a computer system can be measured by the extent of the energy it consumes to complete a task or deliver a piece of work. In other words, it is measured in performance by the total dissipated watts of energy consumption (performance/watts). The performance metric itself can be measured based on the service type whether it is a SaaS, PaaS, or an IaaS. For instance, the performance, in SaaS, can be measured by the number of users requests completed per second. In PaaS, based on the functionality of the software stack of the platform, the performance can be indicated by compilation speed. In terms of the IaaS, the performance can be measured by the utilisation of the processor that runs the tasks. Thus, it is important to make measurement from different aspects in order to provide an overall energy efficiency of a computer system. So, in order to make such energy efficiency improvements, it would be by increasing the performance and maintaining the same energy consumption, maintaining the same performance but with less energy consumption, or more favourably by increasing the performance and reducing the energy consumption at the same time.

Djemame et al [118], [119] emphasised the importance of optimising the energy efficiency at different layers of the Cloud stack and proposed an architecture that supports energy efficiency when constructing, deploying, and operating Cloud services through dynamic intra-layer self-adaptation. Considering the energy consumption has become an essential factor to design and optimise operations to be more energy efficient [78]. Hence, monitoring and profiling as well as forecasting the energy consumption is a key step towards enhancing and optimising the energy efficiency in the Cloud paradigm.

Thus, it is important to model the energy and introduce energy profiling techniques to make awareness about the energy usage of the physical and virtual resources in Cloud environments during the service operation time. The service providers can then make use of these techniques to get energy information and make energy efficient resource management accordingly and avoid ending up with hot spots [8], which may lead to performance degradation and also to increased energy and financial costs to add more cooling systems. Also, the Cloud application analysts and developers can make use of this energy information to design and write energy efficient code and specify energy goals and make their application adapt accordingly while operating. Further, prediction techniques to provide energy information during the service deployment time can be also useful for the Cloud service providers to make energy efficient deployment of the Cloud services. Next, existing energy profiling at both physical and virtual levels will be reviewed, and then followed by a discussion about existing forecasting models for predicting future energy usage.

3.3.2 Physical Machine Profiling Models

There are a number of work focused on analysing and modelling the energy usage in Cloud environments at the PM level. The energy consumption of PMs can be identified and profiled using hardware tools or software tools based on run-time metrics integrated with analytical power models [8].

3.3.2.1 Hardware-Based Energy Profiling

In terms of the hardware tools, the energy consumption of PMs can be easily measured using any of the on-the-shelf wall power meters, like WattsUp meter [120], EnerGenie meter [121], and Kill A Watt meter [122]. The Power Supply Units (PSU) of the PMs get the power via these attached meters. They can measure the aggregated run-time power consumption of a PM, and the measurements can be obtained via USB interface. In terms of the accuracy, these meters can give a measurement accuracy of +/- 1.5% for WattsUp meter, +/- 2% for EnerGenie meter, and +/- 0.2% for Kill A Watt meter.

3.3.2.2 Software-Based Energy Profiling

In terms of the software tools, there are many research works that have investigated how to model the energy of physical machines in order to estimate their energy usage without the use of any hardware tools. Some work focus on estimating energy consumption based on their relation with the utilisations of a number of resource components [123]–[125], and others based only on the CPU resource utilisation [4], [126]–[129].

Resource Usage-Based Energy Profiling

For the sake of reducing the energy consumption in data centres, some works in the literature have investigated the estimation of energy by modelling the energy usage at the hardware-component level within a PM. For example, Kansal et al [123] introduced an additive power model of PMs that considers the idle physical power as a static power and the dynamic power consumed by the physical resources, CPU, memory, and disk, when being utilised based on linear regression models for each component. So, the total dynamic power model is the sum of the power consumed by these physical resource components.

Similarly, Castañé et al [124] introduced a framework, *E-mc2*, that models the energy consumption at fine-grain level of the physical resources in Cloud Computing environments. In their work, they introduced energy models to estimate the energy consumption of internal hardware components, including CPU, memory, network, and disk, and then introduced an aggregate energy model that sums up the energy of these internal components to identify the total energy for each PM.

As stated by Basmadjian et al [125], most of the work estimates the power consumption for the dynamic (active) servers, while assuming the idle (not active) servers just have constant energy consumption regardless of their types. Therefore, they argued that energy-aware algorithms should take into account the power consumption estimations for both idle and dynamic servers in order to take the most proper energy-aware decisions because the idle severs can vary in terms of their power consumption. They proposed models to better estimate the power consumption for the idle servers by breaking it down into its constituent components, like processors, memories, disks, power supply units, and fans [125]. Nonetheless, their proposed models are based on the power characteristics of the current technology. To illustrate, as the technologies evolve over time, their equations of the proposed models would need to be revised because the power consumption behaviour of each component would change as well.

CPU Usage-Based Energy Profiling

Notable work by Fan et al. [126] have introduced a framework to estimate the power consumption of servers based on CPU utilisation only and argued with their results that the power consumption correlates well with the CPU usage. As their framework produced accurate predicted results, they also argued that it is not necessary to use more complex signals, like hardware performance counters, to model power usage. Their work also indicates that the activity of other system components, other than CPU, may have either a small effect on power usage or their activities correlate well with the CPU activity, by having indirect effect on power through triggering the CPU. In their work, they introduced two power models, based on linear and non-linear functions of CPU utilisation. For the linear model, the total power consumption of a PM is identified by summing up its idle power and a fraction of its dynamic power based on CPU utilisation. The dynamic power is defined as the maximum power when the host is fully busy minus its idle power. For the empirical non-linear model, they introduced a calibration parameter and set it to 1.4 in order to reduce the square error [126], but this parameter may change in different types of servers and consequently need to be identified empirically for each type. Running a calibration model to identify such parameter can be considered as a disadvantage, as argued in [78]. The linear power model introduced in [126] has gained a large popularity in the literature and been used and followed by many other works, as in [4], [14], [16], [130]–[132]. However, the applicability of using linear power model depends on the characteristic of the physical machines as other works, like in [4], [133], [134], show that the physical machines considered may not necessarily follow a linear relationship between the power consumption and CPU utilisation. Therefore, characterisation of such type of physical machines is important in order to establish and construct accurate power models accordingly.

Zhang et al [4] argued that modelling the energy consumption based on performance counters, which can be queried from chips or OS, would not work appropriately in heterogeneous environments with different servers' characteristics. Different servers would have different performance counters, resulting in more overhead to use an energy model that would capture all these counters especially if the sampling interval is small. Therefore, the authors presented a *Best Fit Energy Prediction Model* (*BFEPM*) that flexibly selects the best model for a given server based on series of equations that consider only CPU utilisation [4].

Lien et al [127] presented a prediction model to estimate the power consumption of streaming media servers in real time based on the monitored CPU utilisation and the servers configuration parameters without the need of additional hardware tools, e.g. attached power meters. In order to get the required parameters for their model, they introduced three methods. Firstly, a filled-manually method that requires the server's parameters of the base power consumption (idle power) and the full load power consumption (max power) to be filled manually. Secondly, a hardware-revised method requires the use of a hardware power meter only for once to obtain the base power and full load power of a given media server. Thirdly, a software-revised method that predicts the base and full load power consumption based on previous data collection from hardware configurations [127]. Once the base and full load power values of servers have been identified, the authors’ proposed power model is then used to estimate the power consumption of the servers based on a linear relation with their CPU utilisation.

Dargie [128] proposed a stochastic model to estimate the power consumption for a multicore processor based on the CPU utilisation workload. In their work, they found out that the relationship between the workload and power is best estimated using a linear function in a dual-core processor and using a quadratic function in a single-core processor.

Garraghan et al [129] analysed and characterised system failures within a real data of a large-scale production environment, Google cluster data [135], and quantified the wasted energy usage as a result of the failures. The energy usage is not presented in the analysed data. Therefore, in order to identify the energy usage, the authors mapped the analysed servers with three types of servers published by the SPECpower benchmark [136] that have similar characteristics. The power characteristic of the three selected servers from this benchmark show a linear relationship between the power consumption and CPU utilisation. Thus, using a linear power model based on the published results of these three servers, the authors identified the wasted energy as an impact of failures.

3.3.3 Virtual Machine Profiling Models

Unlike PMs, VMs’ energy consumption cannot be measured directly as they do not have direct hardware interfaces to plug in any of the wall watts meters. Therefore, their energy information can be indirectly identified via software tools that model the energy consumed by the PMs in which they are hosted [137] with the use of different approaches, like resource usage-based [123], [130], [138]–[140], lookup table-based [134], and performance counters-based [141], [142].

3.3.3.1 Resource Usage-Based Energy Profiling

Kansal et al. [123] introduced *Joulemeter* as a tool for metering the VM power consumption based on linear regression models of physical power usage and physical resources usage, like CPU, memory, and disk, by each VM. In their models, the physical dynamic power is only attributed to each VM based on the physical resources usage when the VMs are running and inducing workload. However, the physical idle power is not attributed to each VM, and instead used separately and added to the sum of all VMs power consumption to obtain the total PM power consumption for evaluation purposes by comparing it with total PM power consumption obtained by a real external hardware power meter. They considered the limitation of their model that may become over fitted and result in more errors over time. As a way to mitigate this issue, they monitor the real measured power of the server obtained by the wall power meter and compare it with the sum of PM idle power and all estimated VMs power. If the estimation exceeds a certain threshold of errors, their models readapt again and the parameters are relearned based on the recently measured data, which may result in more overhead in using their model and some estimation errors till reaching that threshold to be corrected. In addition, they argued that their VM power model can benefit the Cloud providers to offer energy-aware pricing mechanisms to consider the actual cost of energy usage and be more suitably aligned with the pay-as-you-go billing model. When PMs are switched-on and not running any workload, they still consume a considerable amount of idle energy that comes with costs as well. Nonetheless, their VM power model is limited as it does not attribute the PMs idle power fairly to each VM so that actual cost of idle energy can also be included for each VM and considered for such fair and transparent energy-aware pricing mechanism.

Quesnel et al [138] has argued that most of the work to identify the energy of VMs is only based on the dynamic energy of physical resources, and that physical idle energy should be also included when modelling the energy for VMs because the PMs are still switched-on to host these VMs and maintain their status. Therefore, they introduced a model to attribute the idle energy of a PM to the hosted VMs. In order to get the total energy for each VM, they used an energy model to attribute the dynamic energy to each VM based on the CPU utilisation of the VMs. In their proposed model, the PM's idle energy is attributed to the VMs based on the weight of assigned physical resources, memory and CPU, and the utilisation of these resources by each VM. However, if there is only one or a few idle VMs hosted on a PM and these have been assigned only part of the PM's resources, part of the PM's idle energy is attributed to these VMs. In other words, it means that there will be some part of the PM's idle power not being attributed to the VMs [138]. This contradicts with their motivation that the idle energy of the PMs should be also attributed to the hosted VMs because these PMs are only running to maintain the VMs; otherwise, the PMs could be switched off to save the cost of idle energy consumption. Also, in their model of attributing PM's idle power to the hosted VMs, the resources utilisation by the VM is considered; nonetheless, if the VMs start to utilise some of the assigned resources, then they will start to impact on the PM's dynamic power. Therefore, the extent of the resources utilisation by the VMs should be considered only when attributing the PM's dynamic power consumption to the VMs.

CPU Usage-Based Energy Profiling

Zakarya and Gillam [130] introduced a VM energy model by extending Fan et al [126] linear power model of the physical hosts. In order to identify the total power consumption of a given VM, the PM's idle power is shared evenly among the running VMs. Also, a fraction of PM's dynamic power, which is identified by subtracting the PM idle power from the PM max power, is shared to the VMs based on VM CPU utilisation and the fraction of the PM's total CPU allocated to each VM [130]. However, this work is applicable and limited only to the hosts that follow a linear power model because not all physical hosts follow a linear model of their energy usage and CPU utilisation, as shown in [4], [133], [134]. Also, the authors assume all VMs are homogeneous and divide the physical idle energy evenly among the running VMs on a host. Therefore, fair attribution of the host idle power is not considered when heterogeneous VMs are being hosted simultaneously at the same PM.

Kavanagh et al [140] introduced an IaaS energy modeller that distributes the PM's energy consumption to the VMs. In this modeller, the idle energy consumption of a given PM is divided evenly among the VMs running on that PM, and the active PM's energy consumption is divided to the VMs based on a VM CPU utilisation mechanism. Thus, the introduced energy model would work fairly for attributing the PMs energy to homogeneous VMs. Yet, it lacks consideration of attributing both idle and active PM's energy consumption to heterogeneous VMs running simultaneously at the same PM.

3.3.3.2 Lookup Table-Based Energy Profiling

Jiang et al [134] introduced a method, called *VPower*, that estimates the total power consumption of VMs by considering the static and dynamic power consumption, inferred from the PM. The static power for a given VM is identified by evenly dividing the PM's idle power to all VMs hosted on that same PM. The dynamic power for a given VM is identified by using a two dimensional lookup table (LUT) that returns a specific power value based on given CPU utilisation and Last-Level-Cache (LLC) miss rate. The method requires some time for training with direct measurement at the PM level in order to construct the LUT for a given VM when deployed on a PM for the first time. As the method evenly attributes the PM's idle power among the hosted VMs, it is limited only for homogeneous VMs as it would not be fair for heterogeneous VMs hosted on the same PM to have the same attribution of the idle power.

3.3.3.3 Performance Counters-Based Energy Profiling

A research conducted by Chengjian et al. [141] introduced a model to measure the estimated power consumption of VM using performance events counter. This model attributes power consumption to the VMs based performance event counters of the CPU and memory components when they are being utilised during runtime. The authors argued that the results of the proposed model can get on average about 97% accuracy.

Yang et al. [142] introduced an integrated power model, called *iMeter*, that estimates the power consumption of VMs. Their power model is based on performance counters for CPU, memory, disk, cache, process and network components. These performance counters, e.g. page-reads for memory, are selected based on preforming principal component analysis. Then, the power models are derived using support vector regression (SVR) to estimate the power consumption of VMs based on their relationship with the selected performance counters.

However, the use of performance counters may increase the complexity and overhead when modelling the power usage, as argued in [4], [126]. Also, the proposed models in [141], [142] estimate the power usage of the VMs when they are on and actively inducing some loads; when the VMs are not inducing any load, they have no power usage. In other words, these proposed models do not consider attributing the PM's idle power usage to the VMs, which can be considered as a limitation because the idle power usage of a PM could be saved by switching off the PM when not hosting any VM, as argued in [138].

3.3.4 Forecasting Models

Having discussed the existing work for modelling and profiling energy consumption during the service run-time operation in Sections 3.3.2 and 3.3.3, this section discusses the work for forecasting the energy consumption in future time. Providing energy information of the Cloud services ahead of their operation time can be very beneficial for the service providers to make proactive energy efficient deployment and management of the Cloud services accordingly.

The energy consumption in a system is effected by the running workload. As stated in [143], predicting the energy consumption of Cloud applications and VMs about to be deployed and run would require understanding the characteristics of the underlying physical resources, like idle power consumption and variable power under different utilisation of workload, and the projected virtual resources usage. Therefore, in order to forecast the future energy usage, the workload should be also predicted and then translated into energy consumption.

3.3.4.1 Workload Prediction

In terms of workload prediction, uncertainty issues associated with a Cloud environment makes it difficult to do such prediction, like predicting job runtime in future time. Tchernykh et al [144] have emphasised the difficulty of dealing with uncertainty in a Cloud environment especially since its workload can change dramatically over time. The authors have reviewed and classified the uncertainty issues associated with a Cloud environment and discussed some approaches to mitigate them, for example, using stochastic scheduling, load balancing [145], and adaptive and knowledge-free approaches [146]. Also, another approach presented in [147] looked at the historical data of applications to predict the runtime job of similar applications to be executed.

Knowing the workload of the tasks can help to make efficient task placement strategies. As stated by Patel et al [148], most Cloud infrastructure providers currently ask the users to specify and set the required resources for their jobs/tasks so that appropriate resources are allocated accordingly. However, new customers may find it difficult to specify the actual need of the resources to execute their jobs. Therefore, the authors in [148] proposed a new workload estimation approach based on historical clustered tasks with similar patterns. The approach initially places a new task randomly, and then does continuous monitoring and analysis in order to first get enough data about the initial workload pattern. This initial pattern is then used to map the new task with the historical clustered tasks and accordingly predict the workload, the PMs' CPU usage, ahead of time so that proactive and efficient resource management can take place. However, this approach depends on initial placement of task randomly for a period of time till getting enough observations to identify and map the task pattern with previous clustered tasks in order to do the prediction. This would increase the overhead and latency to predict the workload and may make it inappropriate and difficult to do efficient resource management during the initial period.

Zhang et al. [149] proposed a prediction model to match QoS requirements based on the detection of service workload patterns (SWP). Thus, they use a top-down approach, based on QoS-SWP matrix, to enable dynamic reconfiguration and auto-scaling to meet performance requirements in a Cloud environment. The prediction of workload patterns is based on previous execution and monitoring logs of the resources, including server CPU utilisation, network throughput and data storage size. The aim of this work is to predict the workload patterns of the services and match it with the required QoS to enable proactive resource management.

Khan et al [150] introduced a method to predict repeatable patterns of VMs workload variations over time based on Hidden Markov Modelling (HMM). The workload considered in their work is CPU utilisation. The VMs that exhibit repeatable workload patterns are first explored and then grouped using co-clustering technique into groups with correlated workload patterns. The clustered groups with predictable workload patterns are considered in their prediction method. They classify co-clustered groups to have predictable workload patterns when they show a consistent workload pattern over time or correlated behaviour with other co-clustered groups. Their proposed method based on HMM model uses this clustered information in order to predict the future variation of workload for the VMs. The motivation of their work is to help Cloud providers consolidate and provision their resources efficiently to meet the demand. But they neither predicted the energy usage nor explored the impact of their work on energy usage and cost.

Further, other work has predicted future workload in a Cloud environment based on historical time-series data and using ARIMA model [151]–[154]; nonetheless, their objectives do not consider predicting the energy consumption. For example, Calheiros et al [154] introduced a Cloud workload prediction module based on the ARIMA model to proactively and dynamically provision resources. They define their workload as the expected number of requests received by the users, which are then mapped to predict the number of VMs needed to execute users' requests and meet the QoS.

3.3.4.2 Workload and Energy Prediction

Moreover, other work has looked at forecasting models to predict the workload of resources and energy consumption of such workload, all of which can be based on the analysis of historical data, end-user behaviour and/or predefined types and description of submitted tasks. For example, as stated by Farahnakian et al. [155], it is important to predict the future resource usage in order to energy efficiently manage the Cloud infrastructure resources and avoid violating any SLAs. Therefore, the authors introduced a method, called *LiRCUP*, that predicts the short-time future host workload, CPU utilisation, by using linear regression model and based on recent historical workload. The period of the historical workload considered in their method is for the last hour, as of last 12 intervals, in order to predict the future 5 minutes workload as the next interval. The aim of *LiRCUP* is to determine and predict when an over-loaded host, above 85% of CPU utilisation threshold, is likely to happen so that VMs can proactively be migrated to another host to avoid SLA violation prior to reaching the threshold. Also, *LiRCUP* is aimed to detect when an under-loaded host, less than 10% of CPU utilisation threshold, is likely to happen so that they can migrate VMs to another host and put the predicted under-loaded host into a sleep mode sooner to achieve more energy savings. In order to evaluate this work, the authors implemented the proposed method in CloudSim [18] and used power characterisations of two types of servers, HP ProLiant G4, and HP ProLiant G5, published by SPECpower benchmark [136]. The characterisations of these two servers show a linear relationship between the power consumption and the CPU utilisation. Thus, by using this benchmark, they translated the predicted CPU utilisation into power consumption in order to show the energy savings that could be achieved by their method. Yet, this work is focused on predicting the workload and then the energy consumption only at the host level of Cloud environment and not considering the workload and energy prediction for the VMs.

3.3.5 Overall Discussion

Having such tools that would help to identify the energy usage in Cloud environment is essential in order to help the Cloud application software analysts and developers to design and construct applications with energy-awareness consideration. Also, the Cloud service providers can be facilitated with energy-awareness to enhance their decisions to efficiently manage Cloud resources. Identifying the energy consumption at VMs granularity is considered important to make effective VMs consolidation by taking into account not only the resource utilisation but also the energy consumption as well [142].

Section 3.3 has reviewed the related work on modelling and profiling the power consumption during the run-time in Cloud environments at both physical level, as presented in Section 3.3.2, and at the virtual level, as presented in Section 3.3.3. The following Table 3-1. provides a comparison summary of the closely related work on profiling energy consumption for VMs inferred from their hosting PMs energy consumption.

Table 3-1: Summary of Existing VM Energy Profiling Models

Criteria By	Attributing PM's idle power usage? (Mechanism/Resources)	Attributing PM's active power usage? (Mechanism/Resources)	Type of VMs (Heterogeneous/ Homogeneous)
[123]	Not considered.	Yes. All PM's active power is attributed to VMs based on linear models of PM's power and resources usage, like CPU, memory, and disk, by each VM.	Considered homogeneous VMs only.
[138]	Yes. Part of the PM's idle power is attributed to VMs based on the assigned PM resources, memory and CPU, and the utilisation of these resources by each VM. If any of the PM's CPU or memory resources is fully assigned to VMs, then all PM's idle power is attributed to VMs.	Yes. All PM's active power is attributed to VMs based on their CPU utilisation.	Considered heterogeneous and homogeneous VMs for PM's idle power attribution, and only homogeneous VMs for the PM's active power attribution.
[130]	Yes. All PM's idle power is evenly attributed to the running VMs.	Yes. All PM's active power is attributed to VMs based on the allocated physical CPU resources to each VM and the CPU utilisation by each VM.	Considered heterogeneous and homogeneous VMs for active power, but only homogeneous VMs for the idle power.
[140]	Yes. All PM's idle power is evenly attributed to the running VMs.	Yes. All PM's active power is attributed to VMs based on their CPU utilisation.	Considered homogeneous VMs only.
[134]	Yes. All PM's idle power is evenly attributed to the running VMs.	Yes. All PM's active power is attributed to VMs based on a two dimensional-LUT that returns a specific power value based on given CPU utilisation and LLC miss rate by each VM.	Considered homogeneous VMs only.
[141]	Not considered	Yes. All PM's active power is attributed VMs based on performance event counters of CPU and memory components.	Considered homogeneous and heterogeneous VMs for the active power only.
[142]	Not considered.	Yes. All PM's active power is attributed to VMs by using SVR model to estimate the power consumption of VMs based on their relationship with the selected performance counters of CPU, memory, disk, cache, process, and network components.	Considered homogeneous and heterogeneous VMs for the active power only.

In terms of VM energy forecasting models, it would first require predicting their workload, which then can be translated into energy based on their physical resources usage. As discussed in Section 3.3.4, the work in [148]–[154] used models to predict the workload in Cloud environments in order meet the demand, performance requirements and efficiently provision the resources, but not considering the energy consumption and energy efficiency of the resources, as summarised in Table 3-2. Though the work in [155] predicted the workload, CPU utilisation, and translated into energy consumption to consider the energy efficiency aspect of the Cloud resources, their prediction focus is only at the physical hosts to identify under-loaded and over-loaded hosts so VMs can be moved to another host accordingly. Thus, there is still a need to forecast the energy consumption of VMs prior to their deployment.

Table 3-2: Summary of Forecasting Models

Criteria By	Predicting workload? (Type of workload)	Predicting Energy? (Level of prediction)
[148]	Yes. The considered workload is PM CPU utilisation.	Not considered
[149]	Yes. The considered workload is PM CPU utilisation, network throughput, and data storage size.	Not considered
[150]	Yes. The considered workload is VM CPU utilisation.	Not considered
[151]	Yes. The considered workload is PM CPU utilisation.	Not considered
[153]	Yes. The considered workload is PM CPU utilisation and memory usage.	Not considered
[154]	Yes. The considered workload is number of users requests.	Not considered
[155]	Yes. The considered workload is PM CPU utilisation.	Yes. Energy prediction at PM level.

3.4 Summary

This chapter has reviewed the literature on energy efficient Cloud Computing. Firstly, it has presented and discussed different streams of energy-aware computing, including requirements engineering, programming models, energy-aware resource management, energy efficiency metrics and energy-aware pricing. Secondly, it has reviewed different models for enabling energy-aware profiling at the PMs and VMs in Clouds. Besides, existing forecasting models for future workload and energy usage predictions within Clouds have been also reviewed. This chapter has finally concluded with a summary of the closely related work to this thesis.

To enable energy-awareness in a Cloud environment, an energy-aware Cloud system architecture is proposed in this thesis, which will be discussed thoroughly in the subsequent chapter.

Chapter 4 System Architecture

4.1 Overview

This chapter firstly reviews the motivation and requirements towards energy-aware Cloud Computing, as presented in Section 4.2. Then, it introduces the proposed energy-aware system architecture for enabling energy awareness in Cloud environments, which is presented in Section 4.3. It also provides a thorough description of this architecture's main components and their interactions. Definitions and assumptions considered in this thesis are given in Section 4.4. Finally, this chapter concludes by discussing a number of early experiments conducted on an existing real Cloud testbed to validate the ability of the proposed architecture as a concept for enabling energy profiling at both physical and virtual level, as demonstrated in Sections 4.5 and 4.6.

4.2 Energy Awareness in Cloud Computing

4.2.1 Motivation

Energy consumption has been considered as one of the main operational cost factors in Cloud environments [3]. Cloud service providers encounter a challenge to efficiently maintain and lower the energy usage of their resources in order to reduce the operational cost and maximise the profit [129]. Consideration of the energy usage has been highlighted to be critical to the software analysts and developers of Cloud applications and to the service providers in order to enhance the energy efficiency at different levels of the Cloud stack. The software analysts need to be supported with energy-awareness to specify energy goals from the early stages at the requirements engineering level [11]. Also, programming models should incorporate energy information to help the software developers enhance their programming decisions and construct applications that would meet the required energy goals when operating. Further, energy usage along with resource usage information is critical for the service providers as it can enhance their decisions to efficiently deploy and manage the Cloud services with less energy usage and without performance degradation. Identifying the energy usage along with the resource usage per customer can also help the service providers to set transparent pricing for the offered services with consideration of the actual costs of resources as well as energy usage. Thus, considering energy-awareness from different level of the Cloud stack is very critical to drive towards energy efficient Cloud Computing.

4.2.2 Requirements

In order to efficiently manage and optimise the energy consumption in Cloud environments, tools should be first put in place to provide awareness about the energy usage to be used at different layers of Cloud Computing. As presented in Section 2.2.2, existing Cloud Computing architectures, e.g. as shown on Figure 2-2, describe the main layers along with their components, functionalities and interactions. However, there are missing key functionalities in terms of energy-awareness support within these existing architectures. Therefore, an energy-aware Cloud system architecture is needed to support energy-awareness with consideration of the following requirements:

- ***R.1:*** *The infrastructure layer needs to include the functionality of enabling energy-awareness of PMs and VMs during the service operation.* This is needed to fulfil the above motivation by providing energy information as feedback to enhance the decisions-making for the service providers when managing the resources and for the software analysts and developers when specifying energy goals and optimising the applications.

- ***R.2:*** *The platform layer needs to include the functionality of enabling the energy-awareness proactively prior to the service deployment.* This is needed to help the service providers to predict the energy usage of the resources and make energy efficient service deployment accordingly.

4.3 Proposed Energy-aware Cloud Architecture

Enabling energy-awareness in the Cloud paradigm is a key step towards optimising its energy efficiency. As discussed in the previous Chapter 3, the energy consumption can be easily identified and obtained for the PMs by using any of the shelf Watt meters, but it is difficult and not directly identified for the VMs. Therefore, an energy-aware Cloud system architecture is proposed in this thesis. As depicted in Figure 4-1, this proposed architecture follows the standard reference architecture of Cloud Computing, which consists of three main layers, namely the SaaS layer where the service creation takes place, the PaaS layer where the service deployment takes place, and the IaaS layer where the service operation takes place. This architecture abstracts the details of these three layers and mainly focuses on monitoring and profiling energy consumption during the operation of the services to both the PMs and VMs in order to fulfil the first requirement (**R.1**). Also, it focuses on predicting the energy consumption prior to the deployment of the services in order to fulfil the second requirement (**R.2**). Hence, it is aimed at enabling energy-awareness at the deployment and operational levels of the Cloud paradigm.

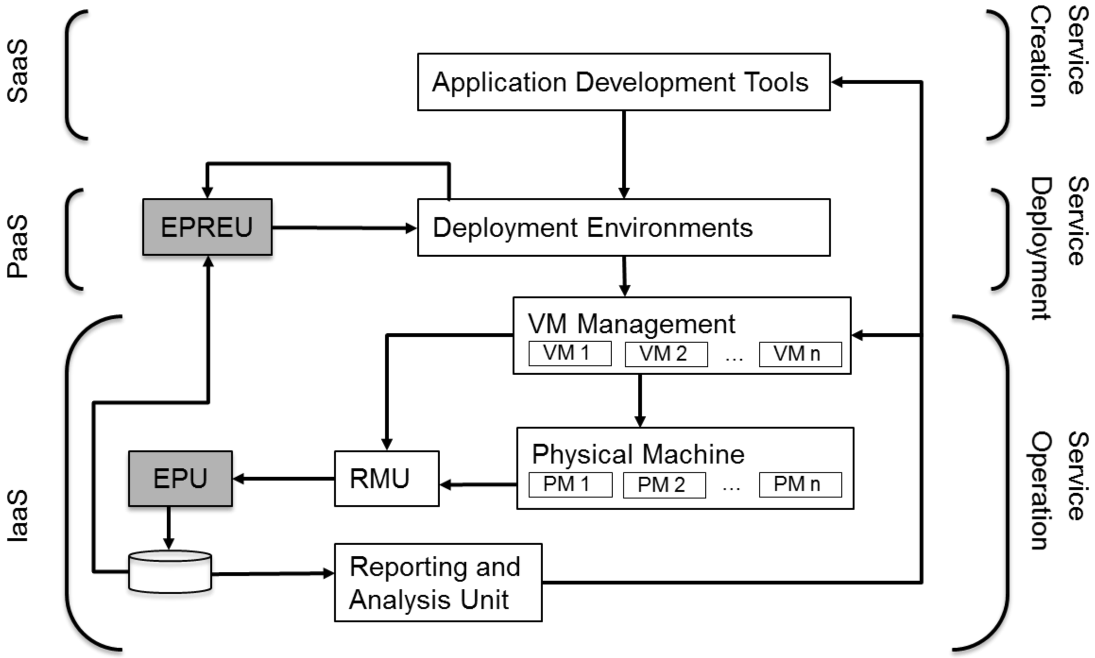

Figure 4-1. Energy-aware Cloud System Architecture

This architecture consists of a number of components, mainly, the Resource Monitoring Unit (RMU), Energy-aware Profiling Unit (EPU), Reporting and Analysis Unit, and Energy-aware PREdiction Unit (EPREU), all of which are discussed in the following Section 4.3.1. The grey highlighted components, EPU, and EPREU, are the key components containing the other two main contributions of this thesis that will be presented in Chapter 5 and Chapter 6, respectively.

4.3.1 Key Components

4.3.1.1 Resources Monitoring Unit

The main role of the RMU is to monitor the resources' usage of the PMs and VMs during the run-time operation, which can be obtained with the use of any of the existing monitoring infrastructure tools, like Zabbix [156]. As the aim of this architecture is for modelling and profiling the energy consumption for Cloud infrastructures, the resources usage to be monitored should be only those that are correlated with the energy consumption. Many of the previous work, as presented in [4], [126]–[128], [130], [140], [155], have used the CPU as the only component when modelling the energy consumption. Therefore, the proposed work follows the same approach and considers the usage of the CPU component to be monitored for PMs and VMs. The energy consumption of the PMs should be also obtained in the RMU, which can be through the use of any of the available wall Watt meters.

4.3.1.2 Energy-aware Profiling Unit

The aim of the EPU is to address the first requirement (**R.1**) by enabling energy-awareness at the VM level. Therefore, an energy-aware profiling model is proposed to identify and profile the energy consumption for the VMs during the operation time. The details of this model will be discussed in Chapter 5.

4.3.1.3 Reporting and Analysis Unit

The Reporting and Analysis Unit is envisaged in this architecture as the tool and link that provides a meaningful feedback, which could be formatted in visualised reports, about the energy usage in the physical and virtual resources of the Cloud infrastructures. This feedback information can be provided to the software analysts and developers in order to enhance their awareness of the energy consumption when specifying energy goals, optimising and constructing the applications. Also, this feedback information can be useful and incorporated by the resource management tools to efficiently provision and manage the Cloud infrastructures resources with energy efficiency in mind.

4.3.1.4 Energy-aware Prediction Unit

The EPREU is aimed to address the second requirement (**R.2**) by forecasting the energy usage of VMs. Thus, an energy-aware prediction framework is proposed to enable forecasting the energy consumption of VMs prior to their deployment. The details of this framework will be discussed in Chapter 6.

4.3.2 Components Interaction

To start with, the Cloud service is created and configured in the application development tool with descriptions of the allocated software and hardware resources. Then, it is deployed in the service deployment environment and goes through VM management at the operational level to run on a given VM hosted by a given PM.

Hence, the proposed system would then start with the RMU to capture and monitor the physical and virtual resources' usage and physical energy consumption along with the number of assigned VMs to each PM during the run-time operation of the Cloud service. Then, EPU, addressing the first requirement (**R.1**), has an appropriate energy model that takes as input the monitored data from RMU and outputs the attribution of the energy consumption to each VM based on the energy consumption of their underlying physical hosts. Next, the EPU profiles and populates these measurements to a knowledge database, which can be further used by the Reporting and Analysis Unit to provide energy-aware reports to the software analysts and developers to help them learn how their applications consume energy and make such energy-efficient decisions accordingly to optimise their applications. Also, these measurements can be very useful for such resource management tools by enhancing their energy-awareness and making energy-efficient decisions when, for example, scheduling the tasks and balancing the workload. Further, this energy-related information of VMs, which can be used by different customers and run on the same PM, can help the service providers introduce a new pricing mechanism that charge the customers based not only on their IT resources usage, but on their energy usage as well.

Moving up to the middle layer when the Cloud services are about to be deployed, EPREU, addressing the second requirement (**R.2**), has a framework consisting of a number of models with the aim of enabling the energy consumption prediction of the requested VMs prior to service deployment. This framework works by considering the type of these VMs and their historical data. The predicted energy consumption for VMs can be incorporated by other deployment strategies to help making energy-efficient decisions proactively.

4.4 Assumptions and Definitions

The following list includes the main assumptions and definitions of variables and terms considered in this thesis:

- ***1:*** The research presented in this research makes abstraction of the type of Cloud applications. Yet, the energy prediction in this research is driven through Cloud application workload patterns; in the essence, it considers Cloud applications having repeated historical workload patterns, static and periodic only, when predicting the energy consumption.
- ***2:*** Power is the rate of electrical usage when performing a work at an instant of time. It can be measured in different units, including Watt (W) and Kilowatt (kW).
- ***3:*** Energy is the averaged power consumption over a period of time to deliver a work. It can be measured in different units, including Watt-Hour (Wh) and Kilowatt-Hour (kWh). Both energy consumption and power

consumption terms have been used interchangeably throughout this thesis.

- ***4:*** VM CPU utilisation represents the workload of the VM when profiling and predicting VM energy consumption at the operational and deployment levels, respectively. It is measured in percentage unit (%).
- ***5:*** VM VCPUs (Virtual CPUs) represents the size of the VM when profiling and predicting VM energy consumption. It is measured as a whole number (e.g. 1, 2, or 3), and it cannot be fractioned (e.g. 0.5 or 1.7).
- ***6:*** Actual ratio of VM VCPUs usage represents the actual usage of the VM VCPUs when predicting VM energy consumption. It is measured as a number between zero and the total number of VCPUs of the VM. It can be fractioned to represent the ratio of actual usage. It is calculated as shown in Equation 4.1 (which uses the above definitions 4 and 5).

$$\boldsymbol{Actual\ Ratio\ of\ VM\ VCPU\ Usage = VM\ VCPUs * \frac{VM\ CPU\ Utilisation}{100}} \quad (4.1)$$

- ***7:*** VM power consumption represents the attributed power consumption of the VM at a given point in time when profiling or predicting. It is measured in W unit.
- ***8:*** PM CPU utilisation represents the workload of the PM when profiling and predicting VM energy consumption. It is measured in percentage unit (%).
- ***9:*** PM power consumption represents the actual or predicted power consumption of the PM at a given point in time when profiling or predicting VM power consumption. It is measured in W unit.

- ***10:*** VM energy consumption represents the attributed energy consumption of the VM over a period of time when profiling and predicting. It is measured in Wh unit.
- ***11:*** PM energy consumption represents the energy consumption of the PM over a period of time. It is measured in Wh unit.
- ***12:*** The term homogeneous VMs is referred to the VMs having the same size in terms of the number of VCPUs (as defined earlier in point 5), e.g. two VMs each with one VCPU.
- ***13:*** The term heterogeneous VMs is referred to the VMs having different sizes based on their number of VCPUs (as defined in point 5), e.g. two VMs, one VM with one VCPU and the other VM with two or more VCPUs.

4.5 Early Implementation

In order to get an early evaluation of the proposed energy-aware Cloud system architecture as a concept for enabling energy-awareness, a number of experiments have been conducted on an existing real Cloud environment, Leeds Cloud testbed. The details of this testbed and how it supports energy-awareness at physical host and VM levels will be discussed next.

4.5.1 Cloud Testbed

The Leeds Cloud testbed consists of a cluster of commodity Dell servers, and each one of these servers has Centos version 6.6 installed as its operating system (OS). Three of these servers with a four core X3430 Intel Xeon CPU on each have been used for the experiments presented in Section 4.6. Also, each server has a total of 8GB of RAM and 250GB of SATA HDD. Additionally, the testbed has a Network File System (NFS) share running on the head node of the cluster and providing a 2TB total storage for VM images.

The architecture of this testbed is shown in Figure 4-2. The testbed utilises OpenNebula [38] version 3.8 as the Virtual Infrastructure Manager (VIM). For the Virtual Machine Monitor or Manager (VMM), the testbed uses Xen [60] hypervisor version 4.0.1 along with the Linux Kernel version 2.6.32.24.

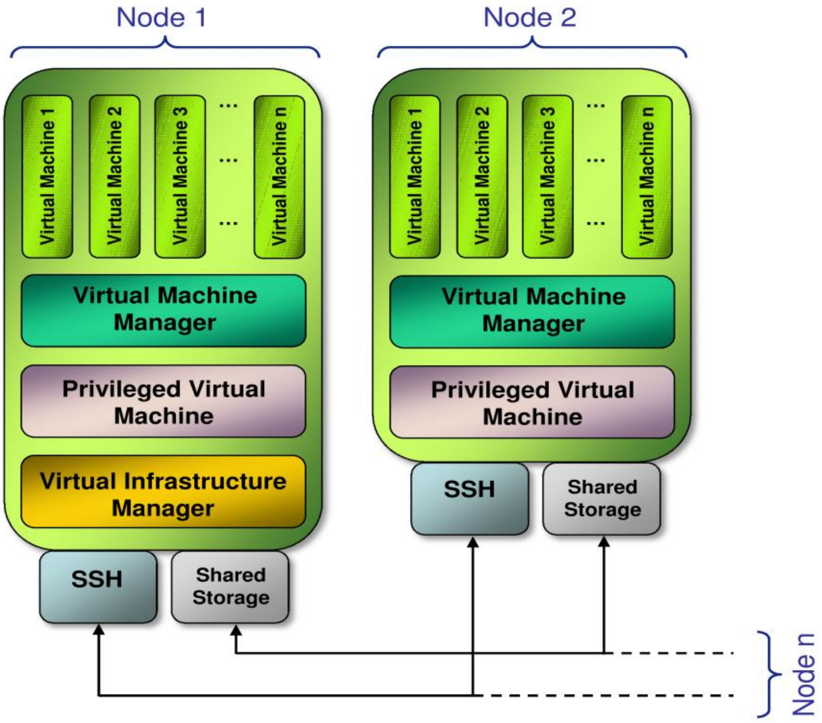

Figure 4-2: Leeds Cloud Testbed Architecture [21]

4.5.2 Monitoring Infrastructure

The resources usage and energy monitoring on the Leeds Cloud testbed is depicted on Figure 4-3. At the physical host level, each of the PM has a WattsUp [120] meter attached via USB interface. These WattsUp meters directly measure power consumption at per second basis for each PM. The measured power values are then pushed to Zabbix [156], which is the monitoring infrastructure tool used in this testbed. Additionally, Zabbix also monitors the resources usage, like CPU, memory and disk, for each of the running PMs and VMs. Finally, the PMs power usage along with the CPU resource usage are sent to the energy-aware profiling unit, which is responsible for enabling energy-awareness at the VM level. The details of the model used within the energy-aware profiling unit will be presented in Chapter 5.

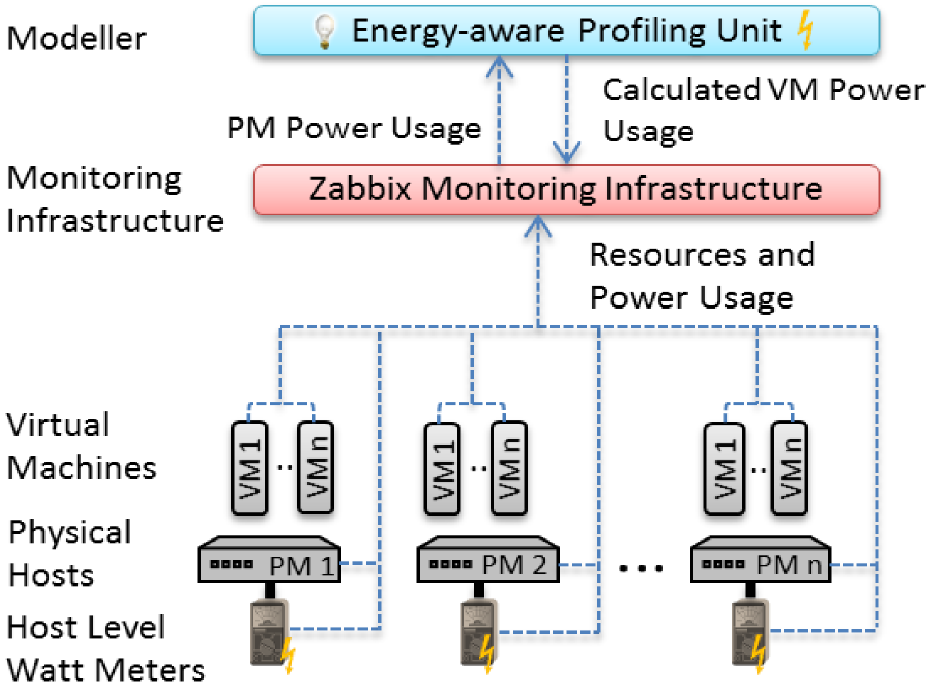
Modeller
Energy-aware Profiling Unit
PM Power Usage
Calculated VM Power Usage
Monitoring Infrastructure
Zabbix Monitoring Infrastructure
Resources and Power Usage
Virtual Machines
VM 1
VM n
Physical Hosts
PM 1
PM 2
PM n
Host Level Watt Meters

Figure 4-3: Monitoring on Leeds Cloud Testbed - adapted from [140]

4.5.3 Specifications of PMs and VMs

As explained earlier, the testbed has a cluster of commodity Dell servers, and the following Table 4-1 summarises the specs of the five PMs considered in this thesis. Hosts A and B are considered in the experiments conducted in Chapters 5 and 6. Hosts C, D and E are considered in the experiments conducted in this Chapter.

The testbed has been upgraded to a newer version, as to be described in Section 5.3, and the earlier version of the testbed has been described before in Section 4.5.1. Hosts A and B are part of the new upgrade of the testbed (to be discussed in Section 5.3) and therefore used for the later experiments conducted in Chapters 5 and 6. At earlier time of this research, Hosts C, D, and E existed as part of the earlier version of the testbed and therefore used for the experiments conducted in this Chapter.

Table 4-1: Specs of the PMs

PM Name	CPU	Memory	Disk
Host_A	A four core X3430 Intel Xeon CPU (default clock speed of 2.40GHz)	Total of 16GB of RAM (four modules of 4GB DDR3 at 1600MHz)	250GB (Model Number: WDC WD2502ABYS)
Host_B	A eight core E3-1230 V2 Intel Xeon CPU (default clock speed of 3.30GHz)	Total of 16GB of RAM (two modules of 8GB DDR3 at 1600MHz)	1000GB (Model Number: ST1000NM0033)
Host_C	A four core X3430 Intel Xeon CPU (default clock speed of 2.40GHz)	Total of 8GB of RAM (four modules of 2GB DDR3 at 1333MHz)	250GB (Model Number: WD5003ABYS)
Host_D	A four core X3430 Intel Xeon CPU (default clock speed of 2.40GHz)	Total of 8GB of RAM (four modules of 2GB DDR3 at 1333MHz)	250GB (Model Number: WD2502ABYS)
Host_E	A four core X3430 Intel Xeon CPU (default clock speed of 2.40GHz)	Total of 8GB of RAM (four modules of 2GB DDR3 at 1333MHz)	250GB (Model Number: WD2502ABYS)

In terms of the VMs considered in the experiments presented in this thesis, Table 4-2 summarises their specs.

Table 4-2: Specs of the VMs

VM Type	VCPU	Memory	Disk
Small VM	1 VCPU	1GB	10GB
Medium VM	2 VCPUs	1GB	10GB
Large VM	3 VCPUs	1GB	10GB

4.6 Experiments and Evaluation

4.6.1 Design of Experiments

Some direct experiments have been conducted on the Leeds Cloud testbed. The overall aim of these experiments is to evaluate the capability of the energy-aware Cloud system architecture as a concept for enabling energy-awareness within a real Cloud environment at both physical host and VM levels. Small VMs with one VCPU on each have been used in these experiments for consistency and to explore the power consumption of the same types of VMs when being run on different hosts, as to be presented in Experiment 3.

In order to design such experiments, a software testing tool that represents real patterns of Cloud applications is needed. Cloud9, a software testing benchmark, has therefore been setup on the testbed to generate real scale-out workloads. The generated workloads by Cloud9 reflect real Cloud applications patterns [157]. Cloud9 is capable of scheduling a task or set of tasks to run on one or multiple VMs, and these tasks can be configured to run in parallel or in stages after each other [158] to represent real pattern of elastic Cloud application.

The following experiments have been designed differently to show various aspects of Cloud Computing patterns as well as energy-awareness at the PM and VM levels. The first experiment has been designed to explore the implication on power consumption when overprovisioning the number of VCPUs on a single VM, having one VCPU only. The second experiment has been designed to explore the impact on power consumption when scaling-out the number of VMs, each having one VCPUs, on the same PM. The third experiment has been designed to show how the power consumption would be influenced when running the same types of VMs, having one VCPUs each, on three different PMs. In order to get the average mean value of the power consumption and eliminate any anomalies of the results, each experiment has been repeated 10 times [159].

4.6.2 Experiment 1

This experiment is designed to schedule some tasks to run dynamically in four stages scaling-up from one VCPU up to four VCPUs on the same VM on a single host. Each stage is set to run for 60 seconds. The following Figures 4-4 and 4-5 show the results of power consumption at host level.

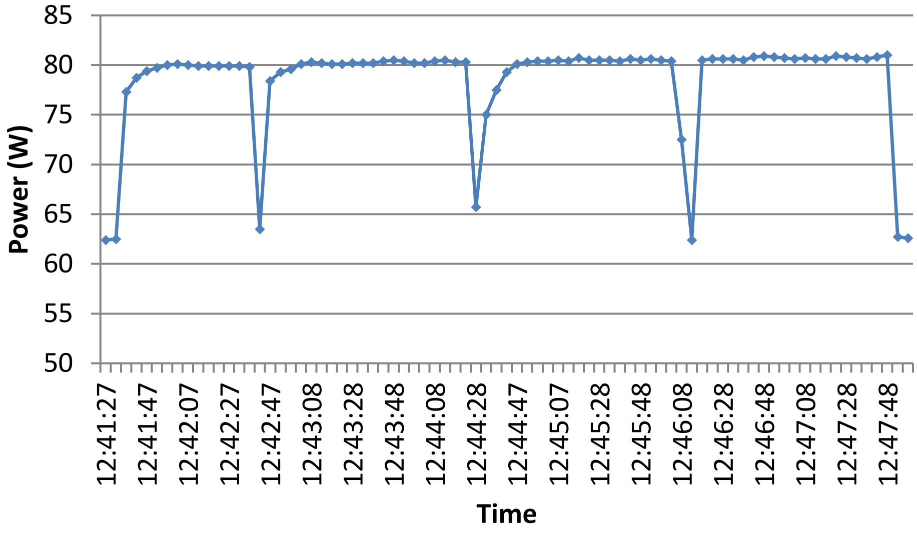

Figure 4-4: VCPU Scaling on a Single VM (Time vs Power)

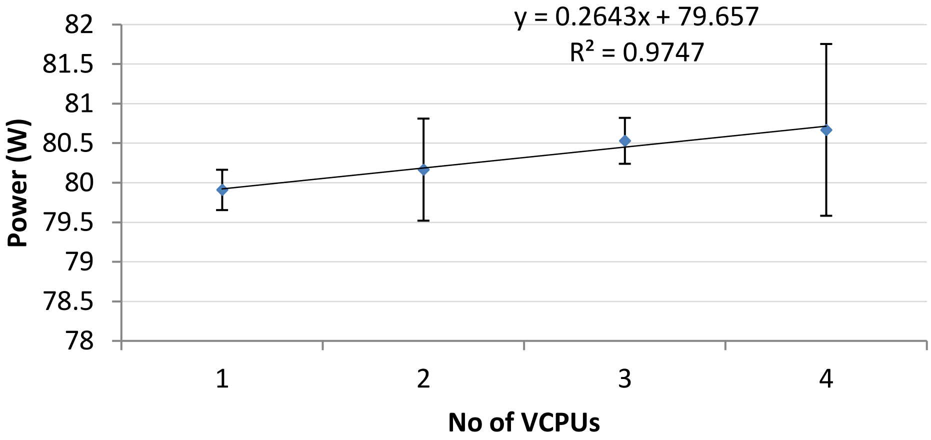

Figure 4-5: VCPU Scaling on a Single VM (No of VCPUs vs Power)

Figure 4-4 shows the results of power consumption for a single run of the experiment, and Figure 4-5 shows the results of the aggregated average of power consumption for each stage over 10 runs. As shown in Figure 4-4, the power consumption at the end of each stage decreases owing to the transition of terminating the current stage and starting the next stage as designed in the Cloud9 benchmark. Each of the four stages shown on Figure 4-4 is set to run only for 60 seconds, but because of the nature of the Cloud9 benchmark, it adds further delays for the transition of starting and ending each stage.

As depicted in Figures 4-4 and 4-5, over-provisioning the number of VCPUs on a single VM does not have an impact on the overall power consumption of the host. The reason in this particular case is that the VM has only one VCPU assigned to one physical CPU. So, overloading that one VCPU with four times of its capacity would still consume the same amount of power. A linear stable trend of the power consumption is represented in Figure 4-5.

4.6.3 Experiment 2

This experiment is scheduled to run some tasks dynamically in four stages scaling-out from one VM up to four VMs on a single host with each stage set to run for 60 seconds.

Figures 4-6 and 4-7 show the results of the power consumption at the host level. Figure 4-6 shows the results of power consumption for a single run, and Figure 4-7 shows the results of the aggregated average of power consumption for each stage over 10 runs.

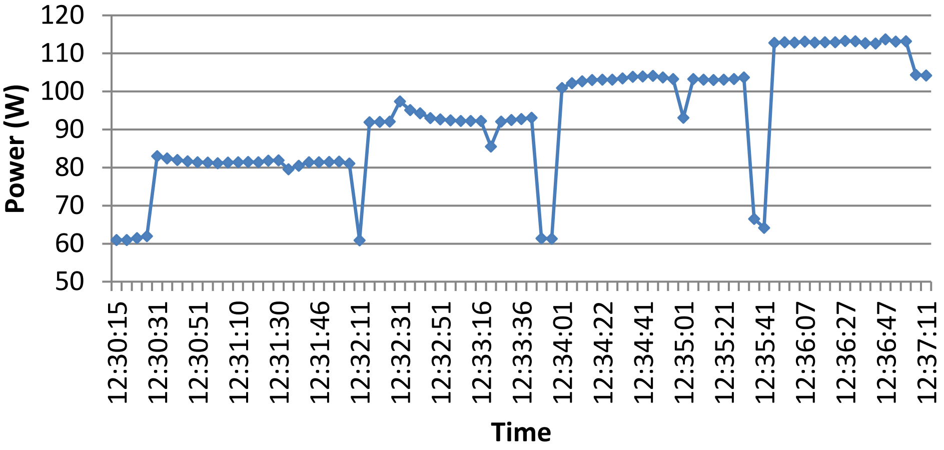

Figure 4-6: VM Scaling on a Single Host (Time vs Power)

As the case with Experiment 1, the transition between each stage results in the reduction of the power consumption, as shown in Figure 4-6. It is clearly shown that increasing the number of VMs from one up to four VMs in a single host has an impact on the overall power consumption for that host. The power consumption shows a linear growth with the increment of VMs. Increasing the number of VMs means increasing the usage of physical resources, like CPU, assigned to these VMs. So, as more physical resources are used, the power consumption increases accordingly.

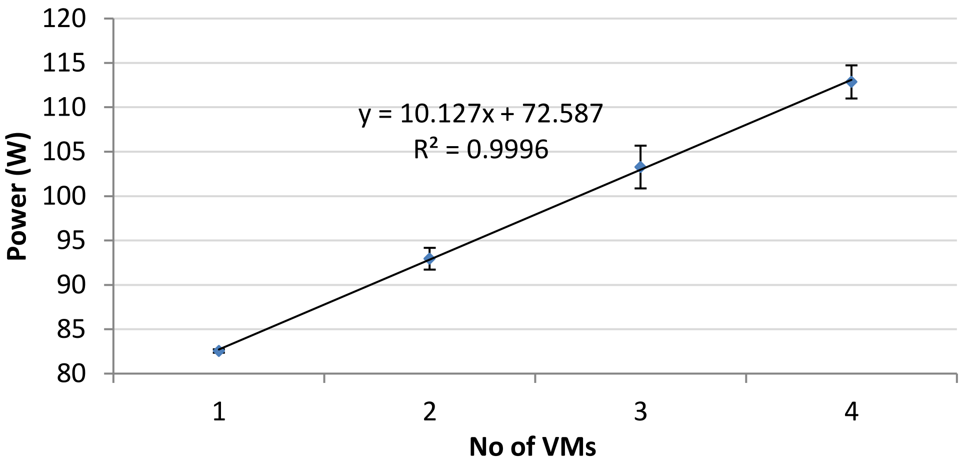

Figure 4-7: VM Scaling on a Single Host (No of VMs vs Power)

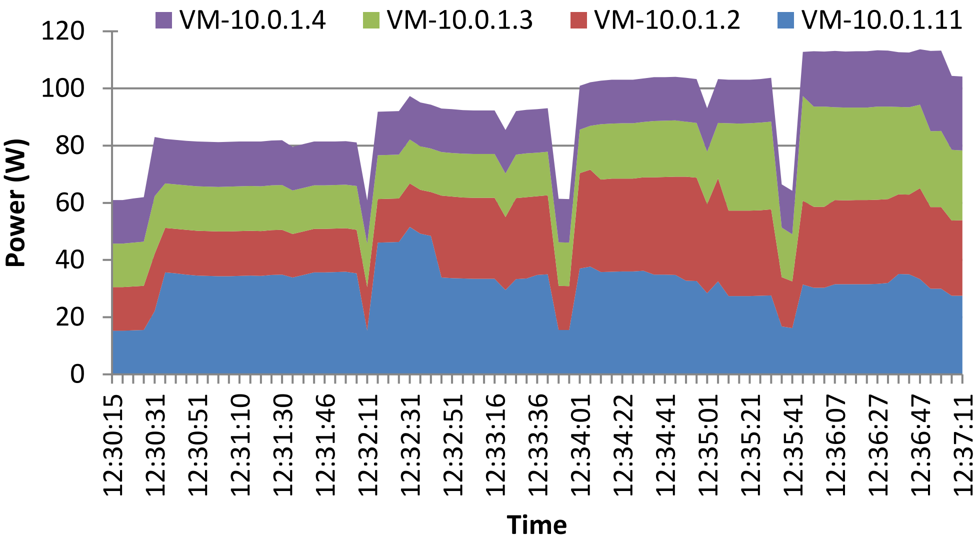

Figure 4-8: VM Scaling on a Single Host (Time vs Power)

Figure 4-8 shows the results of the power consumption at the VM level of the same single run depicted in Figure 4-6. It shows the power consumption for each VM, which has been calculated by using the EPU unit as proposed in the energy-aware Cloud system architecture.

It is clearly shown that total power consumption increases accordingly with the number of VMs used. The total power consumption shown in Figure 4-6 is the same as shown in Figure 4-8; but Figure 4-8 shows the distribution of power consumption among the running VMs on that host thanks to the EPU unit. Before the start of the first stage, all VMs have even distributions of power consumption, but in each stage, the active VMs consume more power than the others in idle state (running but not utilising any workload). This experiment shows that an application consisting a number of tasks can run across multiple VMs simultaneously with energy-aware monitoring and profiling, which can help to identify the energy consumed by an application.

4.6.4 Experiment 3

This experiment has been designed to run some tasks dynamically in three stages scaling-out from one VM up to three VMs across three different hosts simultaneously with each stage set to run for 60 seconds.

Figures 4-9 and 4-10 show the power consumption for each host at physical host level. Figure 4-9 shows the results of power consumption for a single run, and Figure 4-10 shows the results of the aggregated average of power consumption for each stage over 10 runs. Like the previous Experiments 1 and 2, the transition between each stage results in the reduction of power consumption, as shown in Figure 4-9.

Figures 4-9 and 4-10 show that increasing the number of VMs from one up to three VMs across three physical hosts has an impact on the overall power consumption for each host.

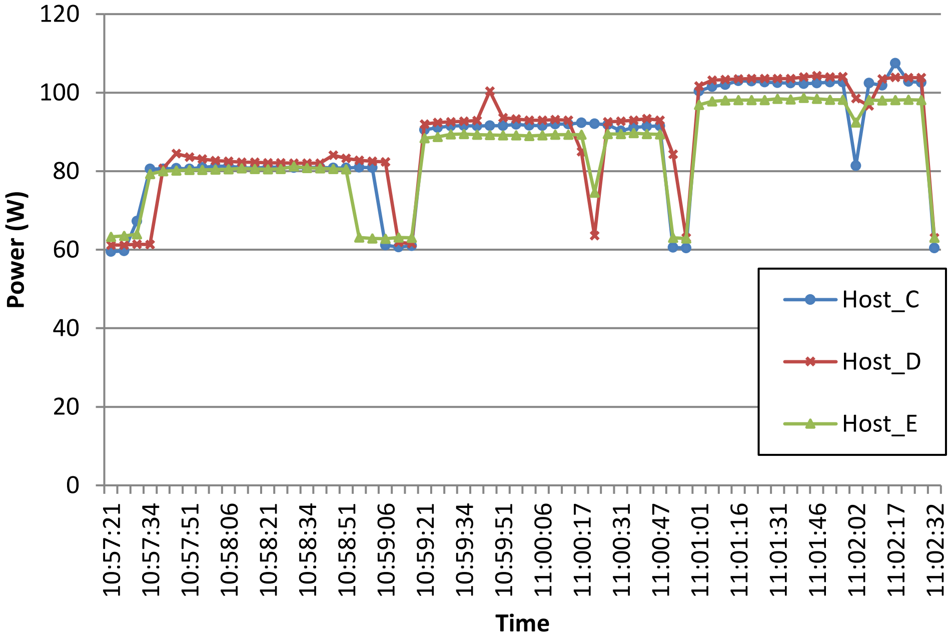

Figure 4-9: VM Scaling on Three Different Host (Time vs Power)

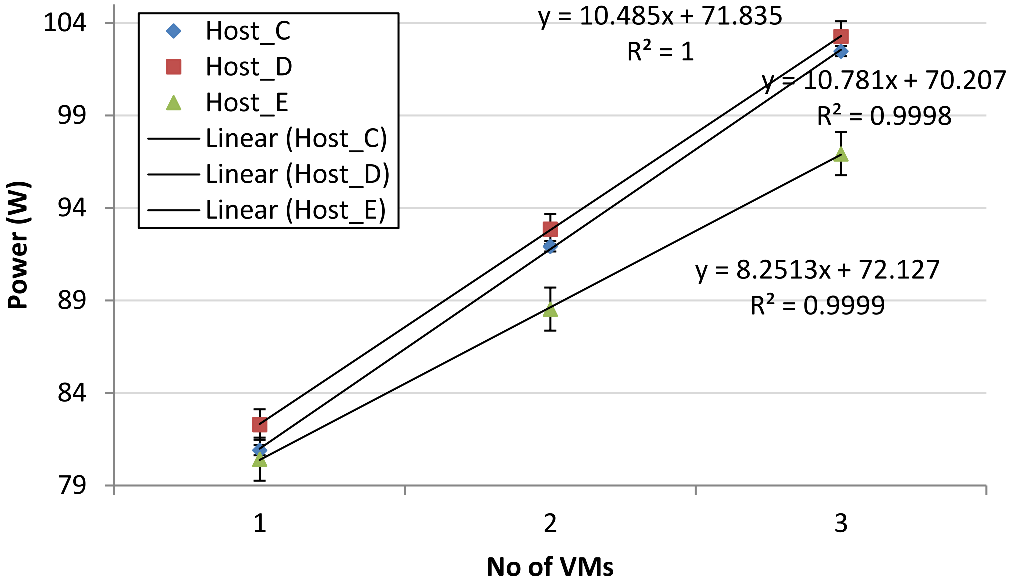

Figure 4-10: VM Scaling on Three Different Host (Time vs Power)

The results shown in this Experiment 3 are similar to those shown in Experiment 2; but here the results are shown for three physical hosts running simultaneously, whereas Experiment 2 for only a single physical host. So, the power consumption in this Experiment 3 increases linearly with the increment of VMs running on each host. This experiment also shows that an application consisting a number of tasks can be scaled-out and run across multiple VMs hosted by different physical host machines at the same time.

4.6.5 Overall Results Discussion

The conducted Experiments 1, 2, and 3 on the testbed have shown an early evaluation of the ability of the proposed energy-aware Cloud system architecture in terms of supporting energy-awareness at the VM level, which addresses the first research question (**Q.1** – see Section 1.3). Now the energy consumption can be identified not only at the PM level, but also at VM level as well.

Also, all of the experiments show that the scalability aspect of Cloud Computing patterns is supported based on the requirements design of the scheduled tasks when running Cloud9. For example, an application consisting of a number of tasks can be scaled-out and run on a number of VMs at the same time on a single or multiple physical hosts with energy-awareness profiling. Hence, this can help to identify the energy usage of an application either running on a single or multiple VMs, which can be certainly useful for the software analysts and developers to monitor and understand the energy usage of their applications. Further, identifying the energy consumption of PMs and VMs can help the service providers to enhance their decisions in order to efficiently manage the resources. For example, as shown in Figure 4-10, Host_E consumes less energy than the other two physical hosts when running two or three VMs. Hence, this information can indicate to the service provider that it is more energy efficient to utilise Host_E fully before utilising the other two hosts.

4.7 Summary

This chapter has introduced the proposed energy-aware Cloud system architecture for supporting energy awareness. The main components of the architecture have been described along with their interactions. A number of experiments have been presented to provide an early evaluation of the ability of the proposed Cloud system architecture as a concept to enable energy-aware profiling at both physical and virtual level. The next chapter will discuss in details the energy-aware profiling model introduced as the key element for facilitating the EPU component of the proposed architecture.

Chapter 5 Energy-Aware Profiling

5.1 Overview

This chapter thoroughly discusses the energy-aware profiling model introduced in this thesis. It firstly discusses on how this model has been developed to identify the energy consumption at the VM level in Cloud environments, as presented in Section 5.2. Direct experiments along with their results are demonstrated to evaluate the capability of the proposed model in terms of fairly attributing the PM's energy consumption to homogeneous and heterogeneous VMs, as presented in Section 5.4.

5.2 Energy-aware Profiling Unit

The aim of the Energy-aware Cloud System architecture introduced in Chapter 4 is to enable energy-awareness in a Cloud environment. The key component to

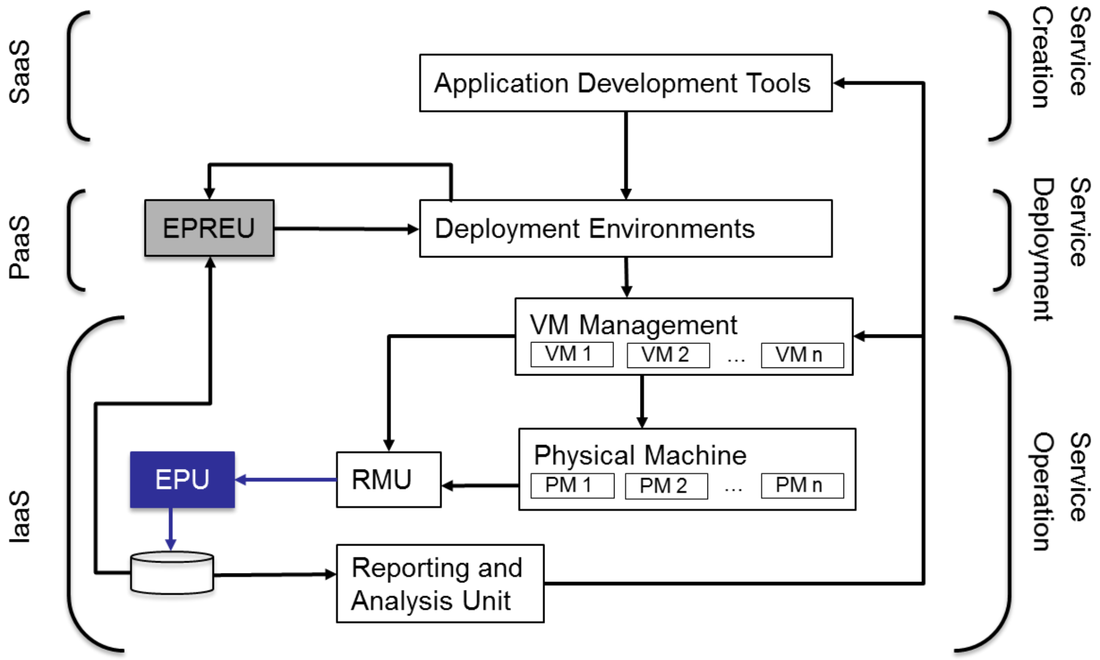

Figure 5-1: Energy-aware Cloud System Architecture - EPU

address and achieve that aim is the EPU unit, as highlighted in blue in Figure 5-1. The objective of EPU is to identify the energy usage of the VMs at the operational level via a mathematical model, which will be introduced in the following Section 5.2.1.

5.2.1 Energy-aware Profiling Model

The power consumption of a PM can be directly measured and mainly consists of two parts, the idle and active power. The idle power is consumed when the PM is turned on but not running any workload. The active power is the extra power induced to the PM when it is busy and running some workload. The total power of the PM can be identified by adding up both its idle and active power.

As the case with the PM, the total power consumption of a VM can be equal to its idle power consumption plus its active power consumption. Yet, the power consumption of VMs is difficult to identify and cannot be directly measured. Hence, the power consumption of VMs can be indirectly inferred from their underlying PMs, which is still challenging and difficult to achieve [123].

A PM can host one or many VMs to run all together at the same time. These VMs can be homogeneous or heterogeneous based on their characteristics, e.g. the number of VCPUs for each VM. Thus, these conditions should be considered when modelling and identifying the power consumption for the VMs.

Previous work has introduced a couple of energy models based on different mechanisms to identify the energy consumption of VMs inferred from their underlying PMs, as discussed earlier in Section 3.3.3. Some of these models, as introduced in [123], [141], [142], just attribute the PMs’ active energy to the VMs. Other models, as introduced in [130], [134], [138], [140], attribute both of the PMs' idle and active energy to the VMs. However, all of these introduced models do not consider a fair attribution the PMs' idle and active energy to homogeneous and heterogeneous VMs running simultaneously.

Therefore, a new energy-aware profiling model is introduced for Cloud infrastructures where the service operation takes place in order to understand how the energy has been consumed at the VM level. The new model aims to overcome the above limitations of the existing VM energy models by not only attributing the PMs' idle and active energy to the VMs, but also fairly attributing the PMs energy to homogeneous and heterogeneous VMs.

Like the power consumption of a PM, the power consumption of a given VM x, VM_{xPwr}, consists of idle and active parts, $VM_{xIdlePwr}$ and $VM_{xActivePwr}$ respectively, as shown in Equation 5.1.

$$VM_{xPwr} = VM_{xIdlePwr} + VM_{xActivePwr} \tag{5.1}$$

The VM idle and active power consumption can be identified based on the idle and active power consumption of the hosting PM. Many of the existing approaches model and identify the energy usage in PMs, as in [4], [126]–[129], and the energy usage in VMs, as in [130], [140], by considering only the CPU resource usage, as presented earlier in Section 3.3. Hence, the work presented in this thesis follows the same approach and considers only the CPU resource usage when modelling and identifying the energy consumption for the VMs.

The new energy-aware profiling model works by fairly attributing the PM's idle energy to VMs based on the number of VCPUs assigned to each VM. As shown in Equation 5.2, $PM_{IdlePwr}$ is the idle power consumption of the PM where the VMs are hosted; VM_{xVCPU} is the number of the VCPUs assigned to the given VM x; VM_{Count} is the number of VMs running on the same PM; and VM_{yVCPU} is the number of VCPUs assigned to a member of the VMs set hosted by the same PM. In this way, the idle energy of the PM is fairly attributed to heterogeneous and homogeneous VMs by considering the size of each VM in terms of the VCPUs assigned to them.

$$VM_{xIdlePwr} = PM_{IdlePwr} \times \frac{VM_{xVCPU}}{\sum_{y=1}^{VM_{Count}} VM_{yVCPU}} \tag{5.2}$$

Also, this model fairly attributes the PM's active energy to the VMs based on the VM CPU utilisation mechanism as well as the number of VCPUs assigned to each VM. As shown in Equation 5.3, PM_{Pwr} is the total power consumption of the PM, from which the PM's idle power is deducted in order to identify the PM's active power; VM_{xUtil} is the CPU utilisation of the given VM x; and VM_{yUtil} is the CPU utilisation of a member of the VMs set hosted by the same PM. This way, the active energy of the PM is fairly attributed to heterogeneous and homogeneous VMs by considering the VM CPU utilisation and number of VCPUs assigned for each VM.

$$VM_{xActivePwr} = (PM_{Pwr} - PM_{IdlePwr}) \times \frac{VM_{xUtil} \times VM_{xVCPU}}{\sum_{y=1}^{VM_{Count}} (VM_{yUtil} \times VM_{yVCPU})} \tag{5.3}$$

Having identified the idle and active power, Equation 5.1 can be replaced with Equation 5.4 to identify the total power consumption for each VM at any given time.

$$\begin{aligned} VM_{xPwr} = PM_{IdlePwr} \times \frac{VM_{xVCPU}}{\sum_{y=1}^{VM_{Count}} VM_{yVCPU}} + (PM_{Pwr} - PM_{IdlePwr}) \\ \times \frac{VM_{xUtil} \times VM_{xVCPU}}{\sum_{y=1}^{VM_{Count}} (VM_{yUtil} \times VM_{yVCPU})} \end{aligned} \tag{5.4}$$

Hence, the introduced energy-aware profiling model can fairly attribute the idle and active energy consumption of a PM to the same or different sizes of VMs in terms of the allocated VCPUs for each VM. For instance, when both a small VM with 1 VCPU and a large VM with 3 VCPUs are being fully utilized on the same PM, the large VM would have triple the value in terms of energy consumption as compared to the small VM; so that the energy consumption can be fairly attributed based on the actual physical CPU resources used by each VM.

In a Cloud environment, a single or part of a physical CPU can be allocated to one or many VCPUs, in which it allows resource overprovisioning. Thus, it is important to consider the right parameter, either the VCPUs or physical CPUs, that would reflect the actual usage of the physical CPU resources. The number of VCPUs parameter is considered in the introduced model because it represents and reflects the actual usage of the physical CPU resources by each VM; this finding has been obtained through empirical direct experiments on a real Cloud environment, as presented in the next section.

5.2.1.1 CPUs Provision to VCPUs

A VM can have one or a number of VCPUs assigned to one or a number of physical CPUs. For the introduced energy-aware profiling model, it is important to know and consider which parameter would reflect the actual usage of physical CPU resources when the VM is utilised. Thus, two experiments have been conducted on the Cloud testbed (see Section 5.3). The aim of these experiments is to identify whether the physical CPU parameter or the VCPU parameter reflects the actual usage of the physical resources and impacts the power consumption. In order to perform a statistical analysis to get the mean values and eliminate any anomalies of the results, each experiment has been repeated five times [140].

Experiment 1: Assigning one physical CPU to one or many VCPUs.

Two VMs have been setup to run on the same PM, which has eight physical CPU cores. The first VM, VM_A, has one VCPU assigned to one physical CPU, and the second VM, VM_B, has three VCPUs assigned to one physical CPU. The experiment has been designed to run at four stages and utilise one VM at each stage by stressing their VCPUs using a tool called Stress [160]. The experiment starts with the first stage utilising 1 VCPU on VM_A, the second stage utilising 1 VCPU on VM_B, the third stage utilising 2 VCPUs on VM_B, and the fourth stage utilising 3 VCPUs on VM_B. Given the fact that each VM has been allocated to one physical CPU only, the aim is to explore the impact on the PM's CPU utilisation and power consumption when these two VMs are being utilised.

Figures 5-2 shows the mean along with the variation of the results over five repeated runs for VM_A CPU utilisation. Recall, this VM_A has only one VCPU assigned to one physical CPU, and this VCPU has been fully utilised only during the first stage. Thus, its CPU utilisation is at 100% during the first stage and idling for the remaining stages.

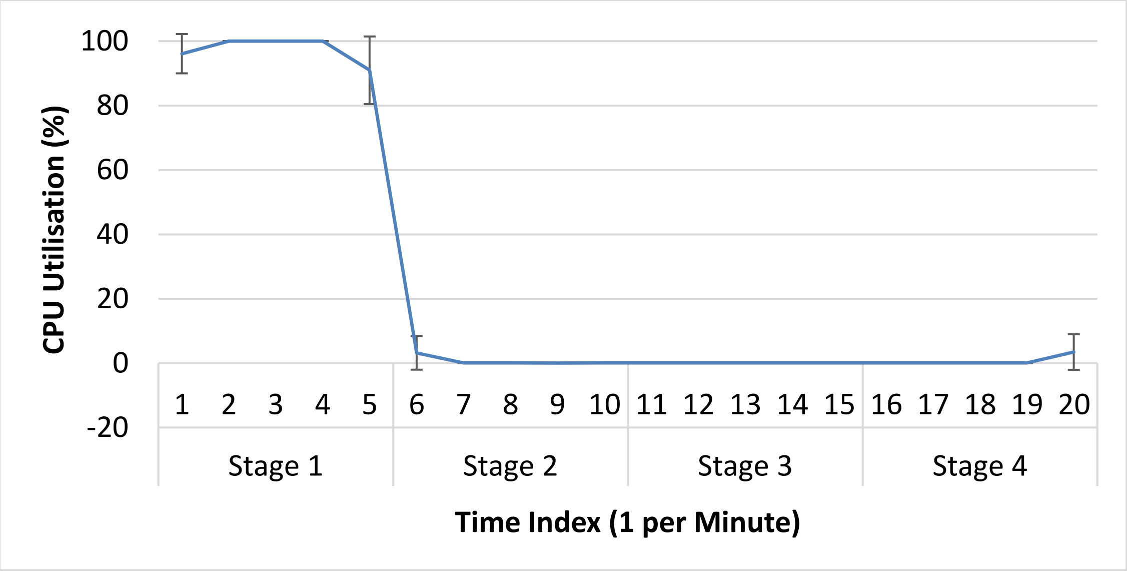

Figure 5-2: Mean CPU Utilisation for VM_A

Figure 5-3 shows the mean along with the variation of the results for VM_B CPU utilisation. This VM_B has three VCPUs assigned to one physical CPU, and it utilises 1 VCPU during stage two, 2 VCPUs during stage three, and 3 VCPUs during stage four. Thus, its CPU utilisation is idling during the first stage, at 33% during the second stage, 66% during the third stage, and 100% during the fourth stage.

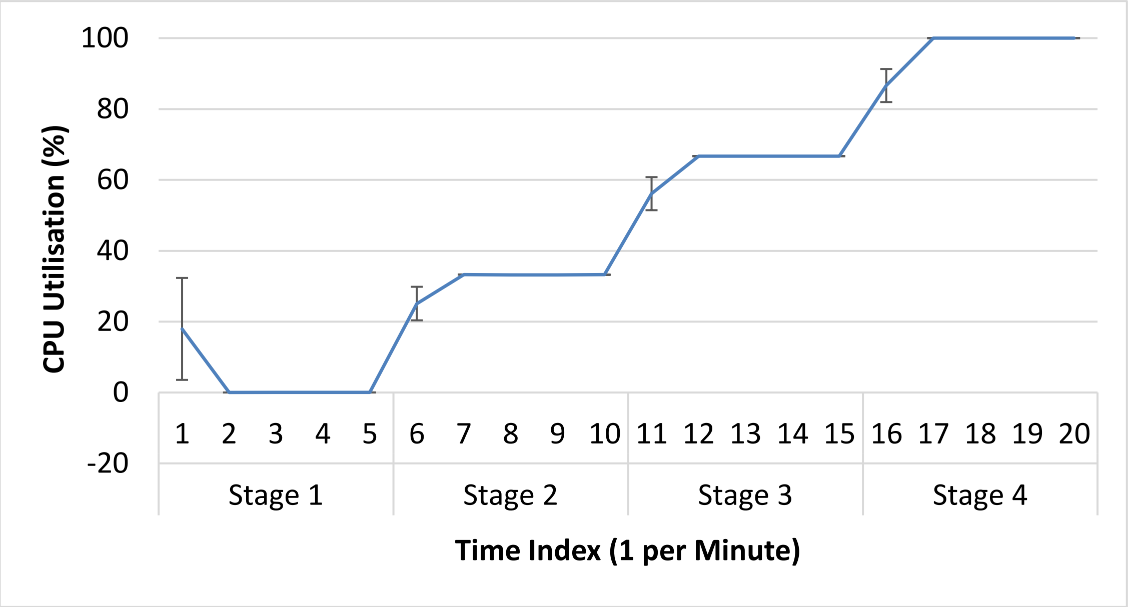

Figure 5-3: Mean CPU Utilisation for VM_B

Figure 5-4 shows the variation along with the mean of CPU utilisation and power consumption for the PM over five runs. These results indicate that the PM's CPU utilisation and power consumption stay the same during the first and second stages, and then increase for each subsequent stages three and four. Thus, even though VM_B has three VCPUs assigned to only one physical CPU, the observed results reveal that VM_B actually uses three physical CPUs when it is utilising all of its VCPUs, as shown during the fourth stage.

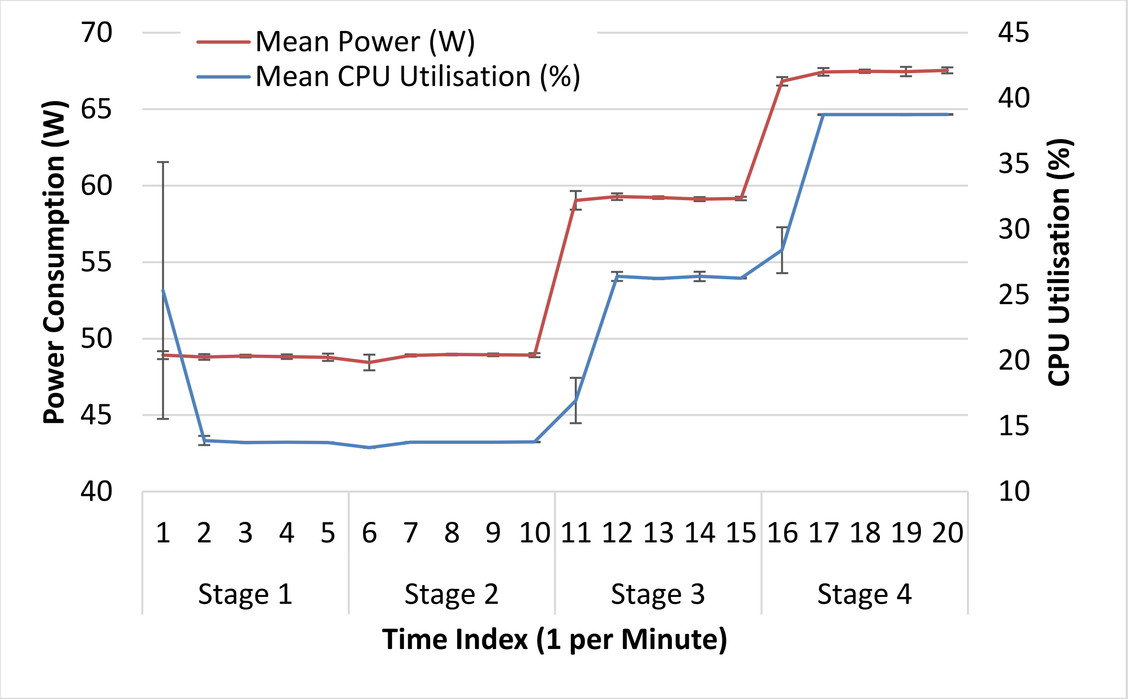

Figure 5-4: Mean Power Consumption and CPU Utilisation for the PM

Experiment 2: Assigning half, one and multiple physical CPUs to one VCPU.

Three VMs have been setup to run on the same PM, which has eight physical CPU cores. The first VM, VM_A, has one VCPU assigned to half of a physical CPU, the second VM, VM_B, has one VCPU assigned to one physical CPU, and the third VM, VM_C has one VCPU assigned to three physical CPUs. The experiment has been designed to run at three stages, starting with the first stage utilising 1 VCPU on VM_A, the second stage utilising 1 VCPU on VM_B, and the third stage utilising 1 VCPU on VM_C. Given the fact that each VM has been assigned with different number of physical CPUs, the aim is to explore the impact on the PM's CPU utilisation and power consumption when these VMs are being utilised.

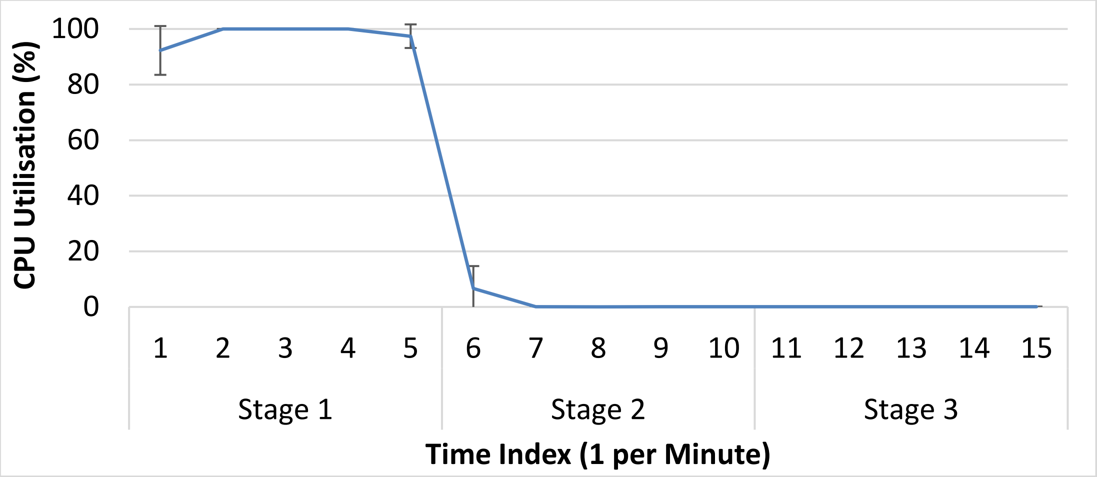

Figure 5-5: Mean CPU Utilisation for VM_A

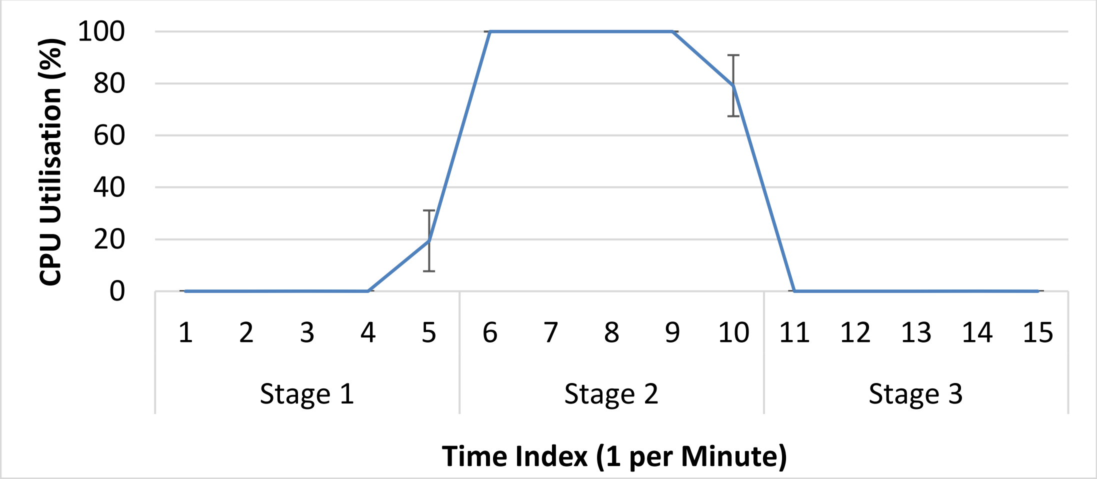

Figure 5-6: Mean CPU Utilisation for VM_B

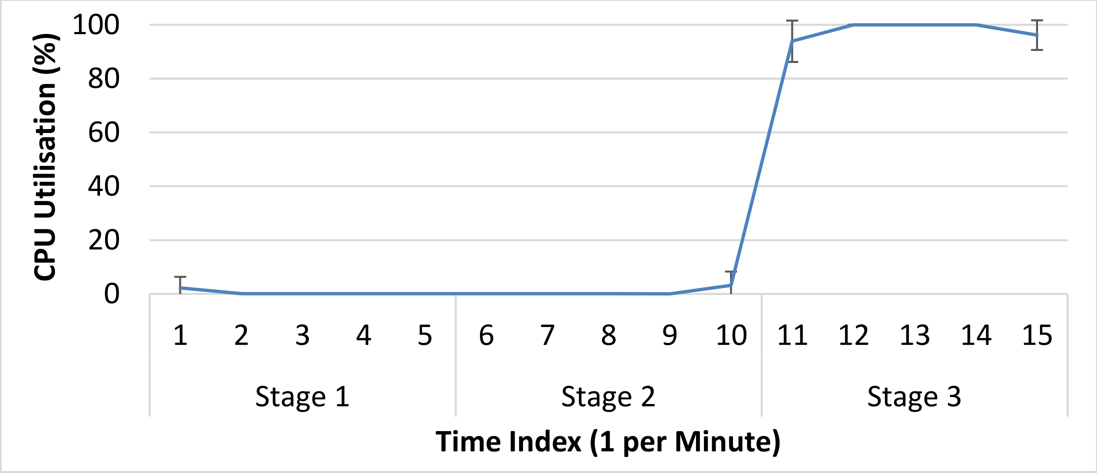

Figure 5-7: Mean CPU Utilisation for VM_C

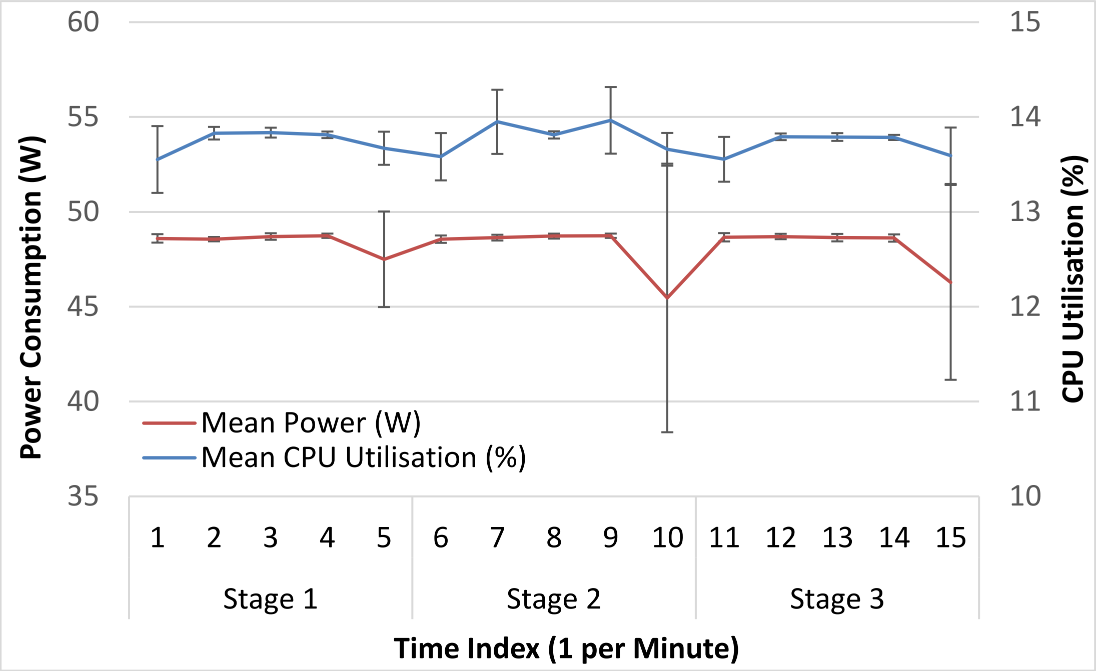

Figure 5-8: Mean Power Consumption and CPU Utilisation for the PM

As designed in this experiment, each one of the three VMs has one VCPU, which has been fully utilised during the first stage for VM_A, second stage for VM_B, and the third stage for VM_C, as shown on Figures 5-5, 5-6, and 5-7. Yet, as each one of the VMs has a different number of physical CPU assignment, it is expected that each VM would have a different impact on the actual usage of physical CPU when being fully utilised.

However, the results on Figure 5-8 reveal that the hosting PM's mean CPU utilisation and power consumption stay the same over the three stages. These results indicate that when each VM has one VCPU and is being fully utilised, it attempts to use one physical CPU regardless of whether it is being allocated half or multiple physical CPUs.

Overall Results Discussion and Finding

The first experiment reveals that even though when a VM has three VCPUs assigned to only one physical CPU, that VM can use up to three physical CPUs when being fully utilised. The main reason of having that is because the default settings of the current VIM, OpenNebula, along with the hypervisor, KVM, on the Cloud testbed support overselling the physical CPU resources. Moreover, there is no such policy implemented in the testbed to enforce hard restrictions on the assignment of physical CPUs to the VMs (VCPUs). Also, the second experiment reveals that when a VM has one VCPU assigned to either a half or many physical CPUs, that VM uses only one physical CPU when being fully utilised.

Therefore, it can be concluded when each VCPU is being utilised on a VM, it will attempt to use one physical CPU regardless of the assignment of the physical CPUs to the VMs. Hence, the main finding of these two experiments is that the VCPU should be considered as the key parameter that would reflect the actual usage of the physical CPU resources by each VM. Nonetheless, it should be noted that in case a hard policy is enforced to limit the physical CPU usage by each VM based on the assigned number of physical CPUs, then the physical CPU parameter should be considered.

5.3 Implementation

In order to evaluate the capability of the introduced energy-aware profiling model to fairly attribute the energy usage for homogeneous and heterogeneous VMs, a number of direct experiments have been conducted on the Cloud testbed (see Section 4.5). This testbed currently supports OpenNebula [41] version 4.10.2 as the VIM, and KVM [161] hypervisor for the VMM.

The power consumption at the PM level is obtained by the WattsUp meter [120]. The power consumption at the VM level is identified by the EPU, which works offline based on the introduced energy-aware profiling model. A profiler facilitating the introduced model has been created in shell script. This profiler works offline by taking as input the CPU resource usage for the VMs along with the power usage for the PM on which the VMs are hosted. It then outputs the profiled power consumption for each VM based on the introduced energy-aware profiling model.

5.4 Experiments and Evaluation

5.4.1 Design of Experiments

The overall aim of the experiments is to demonstrate that the new energy-aware profiling model is capable of fairly attributing the PM's energy consumption to homogeneous and heterogeneous VMs based on their CPU utilisation and size. The size of the VM is identified by its capacity in terms of the number of VCPUs. In the following, if two VMs have the same number of VCPUs on each, then they are considered homogeneous VMs. If one has one VCPU and the other has two or more VCPUs, then they are considered heterogeneous VMs.

The experiments consider three sizes of VMs, a small VM with one VCPU, a medium VM with two VCPUs, and a large VM with three VCPUs. The first experiment is designed to run two small VMs on the same PM to show how the energy consumption is attributed to homogeneous VMs, as to be presented in Section 5.4.2. The second experiment is designed to run a small VM and a large VM on the same PM to show how the energy consumption is attributed to heterogeneous VMs, as to be presented in Section 5.4.3. The third experiment is designed to run a small VM, a medium VM, and a large VM on a PM, and also to run the same types of these three VMs on another PM having different characteristics, as to be demonstrated in Section 5.4.4. The aim of the third experiment is to explore how the energy consumption is attributed to the same types of VMs when being run on different PMs.

In terms of inducing workload on the VMs, *Cloud9* [157], is a suitable software tool to use for fully stressing each VCPU on the VM, as introduced in Section 4.6.1. However, it is not flexible in terms of stressing the VCPUs of the VM at certain utilisation level, e.g. 80%. Therefore, the software tool *Stress* [160] is used along with *cpulimit* to generate workload on the VMs at any level of CPU utilisation. All of the VMs used in these three experiments are designed to be idle for 15 minutes at the first stage, and then actively run at 80% of CPU utilisation for another 15 minutes at the second stage. This way can help to explore how the idle and active power consumption of the PM are attributed to the VMs over time. All of the experiments are repeated five times and the statistical analysis is performed in order to consider the mean values of the results and eliminate any anomalies [140].

5.4.2 Experiment 1: Two Homogeneous VMs on a Single Host

This experiment shows the results of attributing the power consumption to two homogeneous small VMs, VM_A and VM_B, running on the same PM. The mean power consumption and CPU utilisation for VM_A and VM_B are shown in Figures 5-9 and 5-10, respectively. Recall, both VMs are idle in the first stage during the first 15 minutes and active in the second stage with 80% of CPU utilisation during the remaining 15 minutes. The vertical error bars illustrate the standard deviation, which is very small and not noticeable for the CPU utilisation and the power consumption during the first stage.

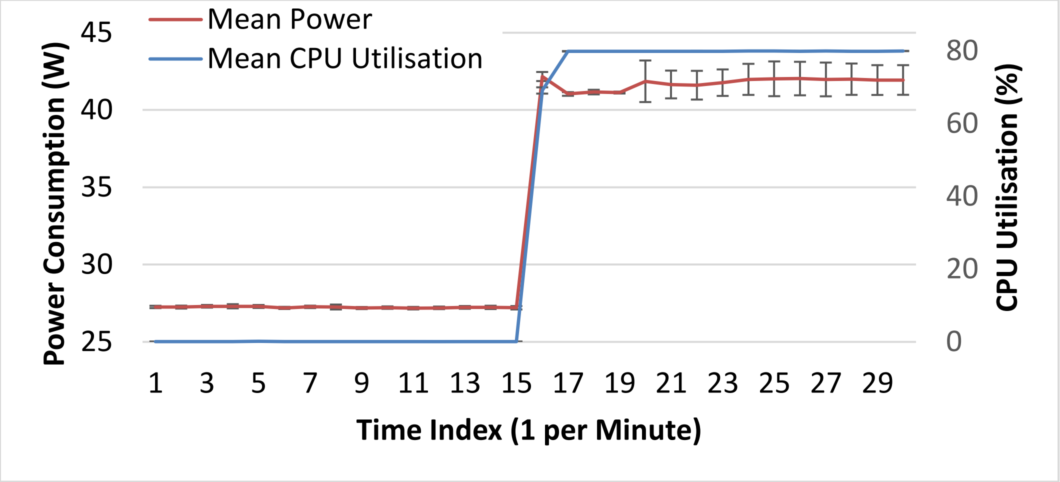

Figure 5-9: Mean Power Consumption and CPU Utilisation for VM_A

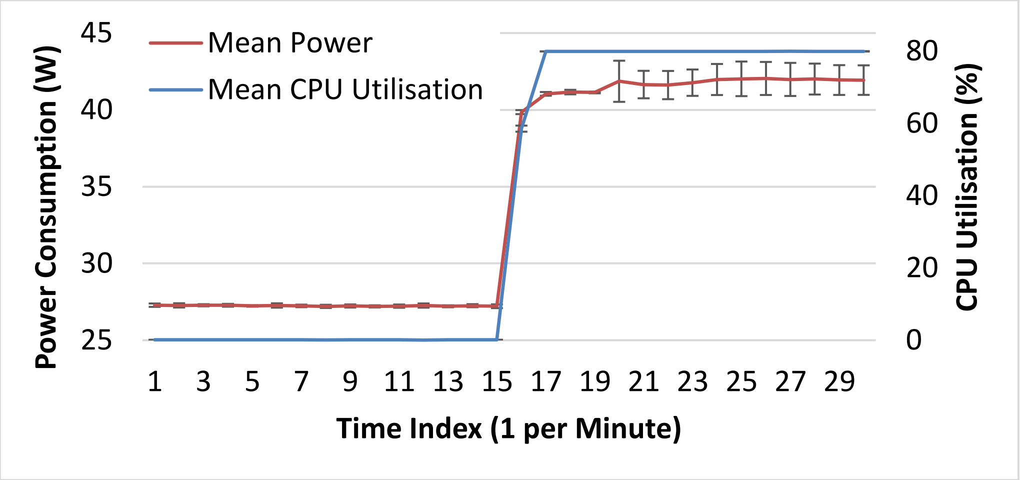

Figure 5-10: Mean Power Consumption and CPU Utilisation for VM_B

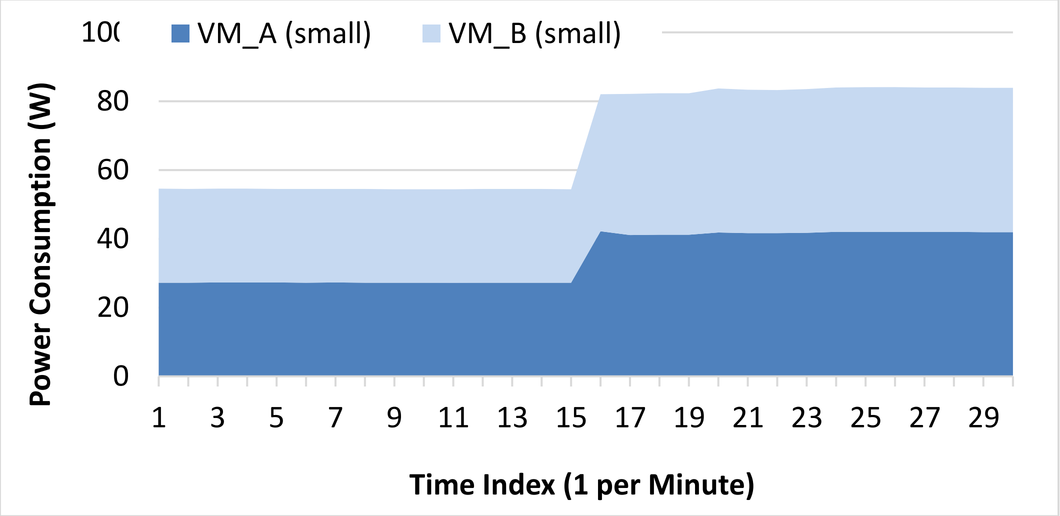

Figure 5-11: PM Mean Power Consumption Attributed to each VM

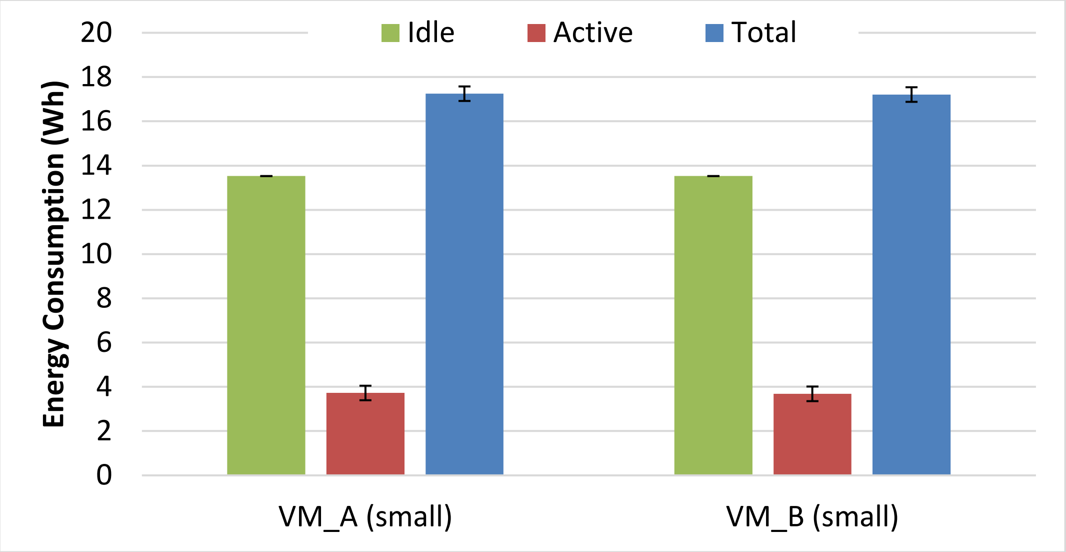

Figure 5-12: Mean Energy Consumption per VM (for 30 minutes)

Figure 5-11 shows the distribution of the PM's mean power consumption to these two VMs, and Figure 5-12 shows the mean energy consumption per VM. As these two VMs are homogeneous and run the same workload, they have the same attribution of the PM's idle and active energy consumption. Hence, the energy-aware profiling model is capable of fairly attributing the PM's energy consumption to homogeneous VMs.

5.4.3 Experiment 2: Two Heterogeneous VMs on a Single Host

This experiment shows the results of attributing the energy consumption to two heterogeneous VMs, VM_A (small VM) and VM_B (large VM), running on the same PM. The mean power consumption and CPU utilisation for VM_A and VM_B are shown in Figures 5-13 and 5-14, respectively. Recall, both VMs are idling for the first 15 minutes and actively running with 80% of CPU utilisation for the remaining 15 minutes. Similarly, the vertical error bars are used to illustrate the standard deviation, which is very small and not clearly noticeable.

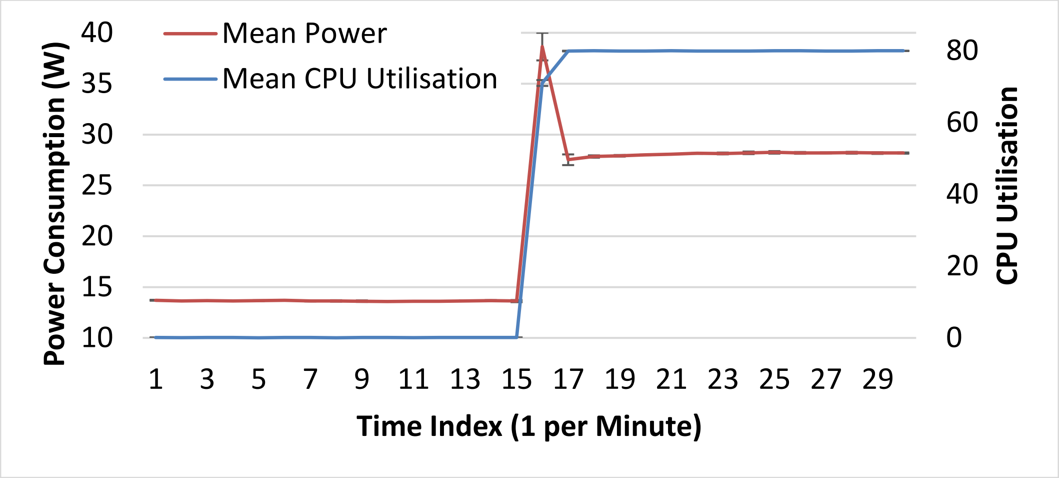

Figure 5-13: Mean Power Consumption and CPU Utilisation for VM_A

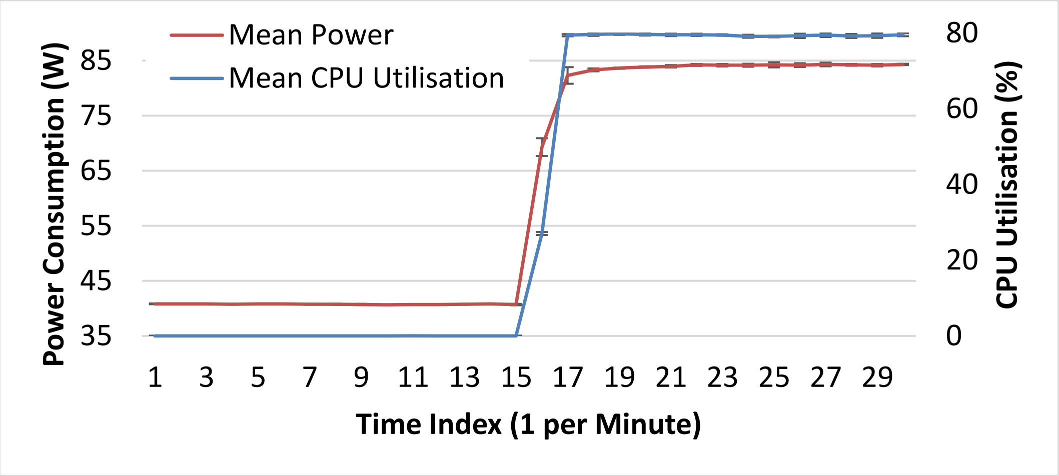

Figure 5-14: Mean Power Consumption and CPU Utilisation for VM_B

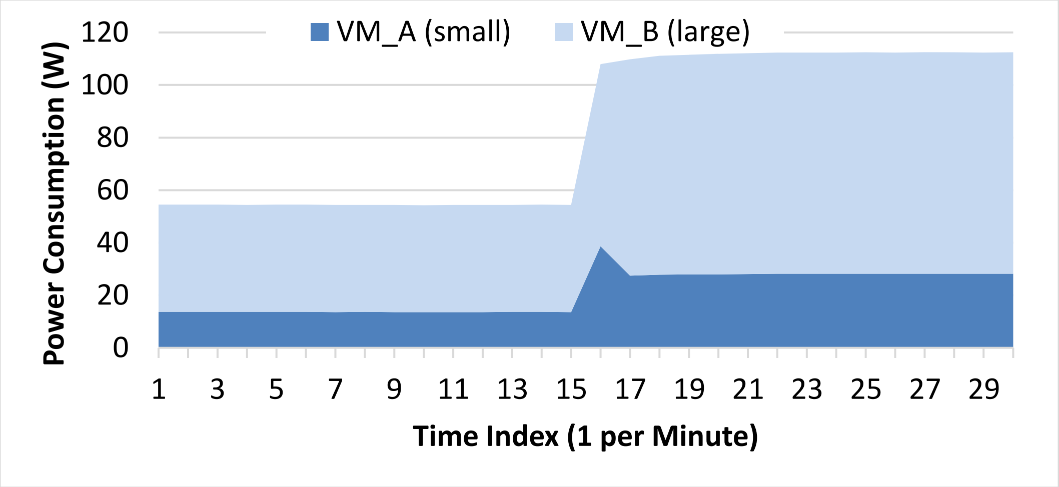

Figure 5-15: PM Mean Power Consumption Attributed to each VM

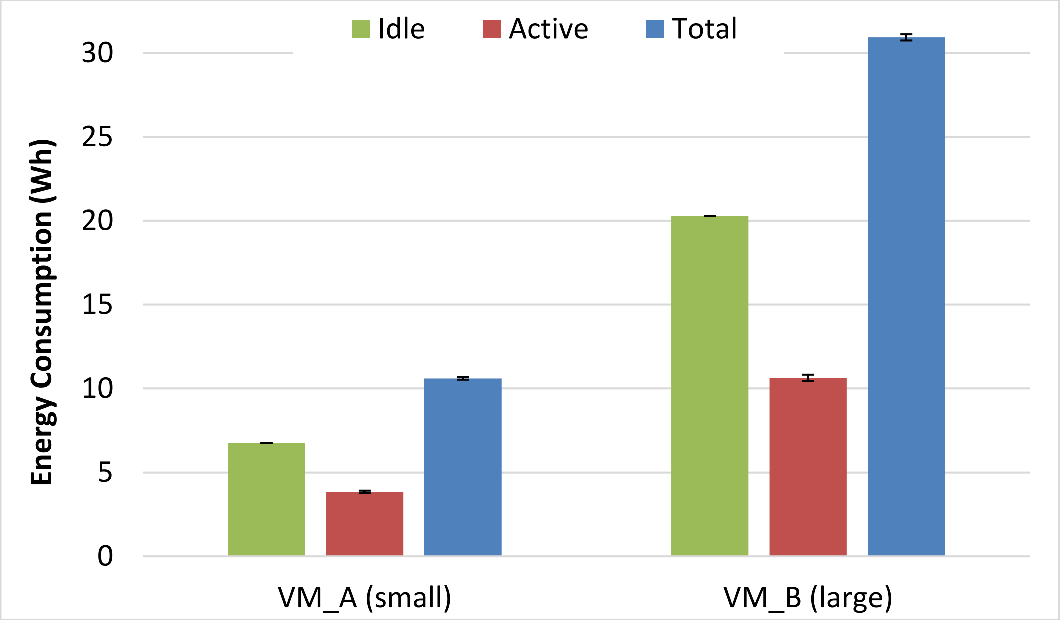

Figure 5-16: Mean Energy Consumption per VM (for 30 minutes)

Figure 5-15 shows the distribution of the PM's mean power consumption to these two VMs over time, and Figure 5-16 shows the mean energy consumption per VM. Both VMs run at 80% of CPU utilisation, but as they are heterogeneous they have different attribution of the idle and active energy consumption, which fairly corresponds to their size. As having triple the size in terms of VCPUs, the energy consumption of the large VM, VM_B, during the idle and active stages is about three times larger than the energy consumption of the small VM, VM_A. Overall, the results show that the energy-aware profiling model is capable of fairly attributing the PM's energy consumption to heterogeneous VMs based on their utilisation and size, which reflect the actual physical resources' usage.

5.4.4 Experiment 3: Heterogeneous VMs on Different Hosts

This experiment shows the results of energy consumption attribution to three heterogeneous VMs, VM_A (small VM), VM_B (medium VM), and VM_C (large VM) running on a PM, Host_A. Additionally, this experiment also presents the results of attributing the same types of these three VMs on another PM, Host_B.

5.4.4.1 Host A

The mean power consumption and CPU utilisation for VM_A, VM_B and VM_C running on Host_A are shown in Figures 5-17, 5-18 and 5-19, respectively. As designed, all of the VMs are idling for the first 15 minutes and actively running with 80% of CPU utilisation for the remaining 15 minutes.

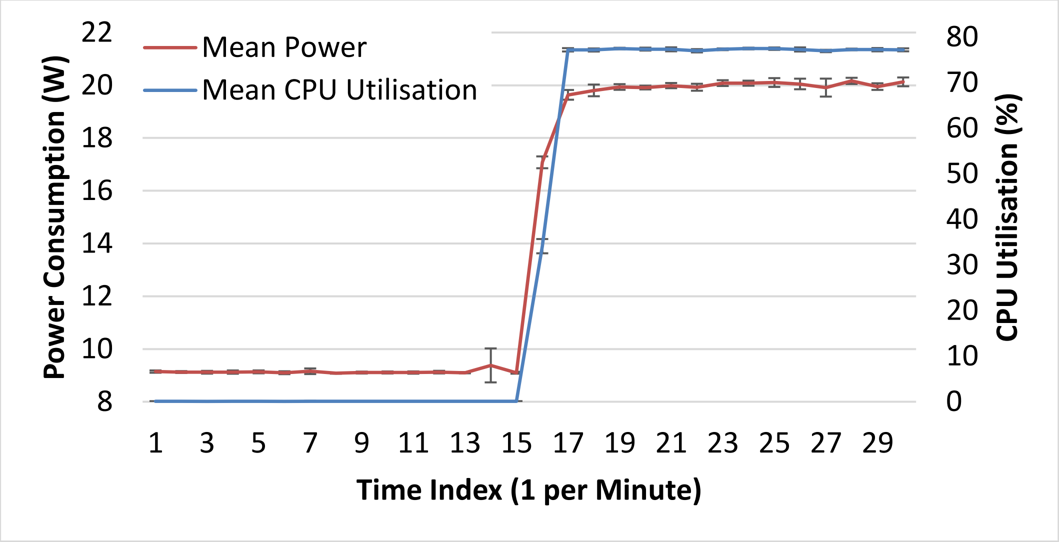

Figure 5-17: Mean Power Consumption and CPU Utilisation for VM_A

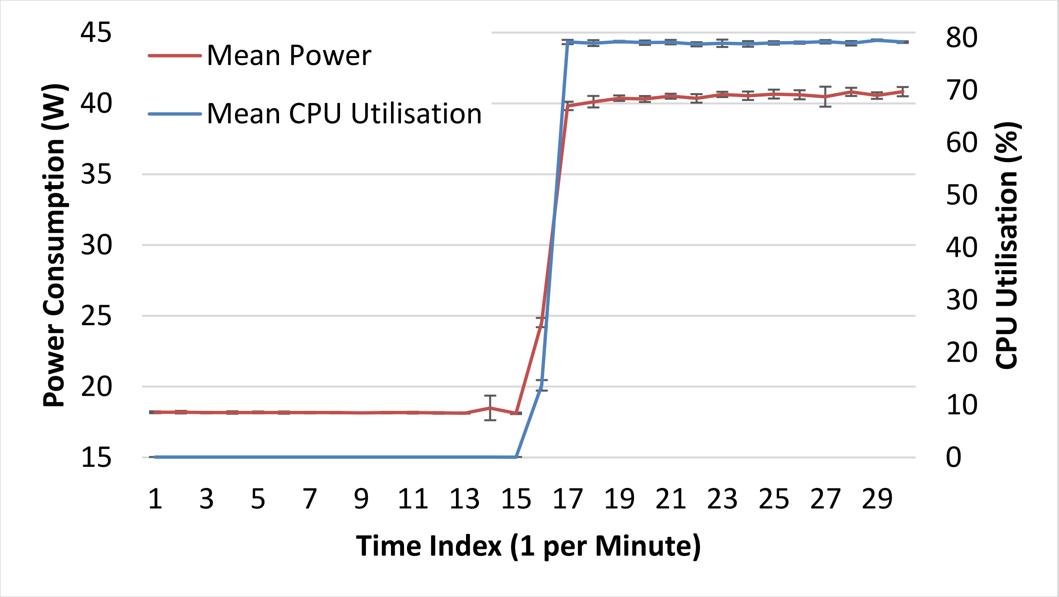

Figure 5-18: Mean Power Consumption and CPU Utilisation for VM_B

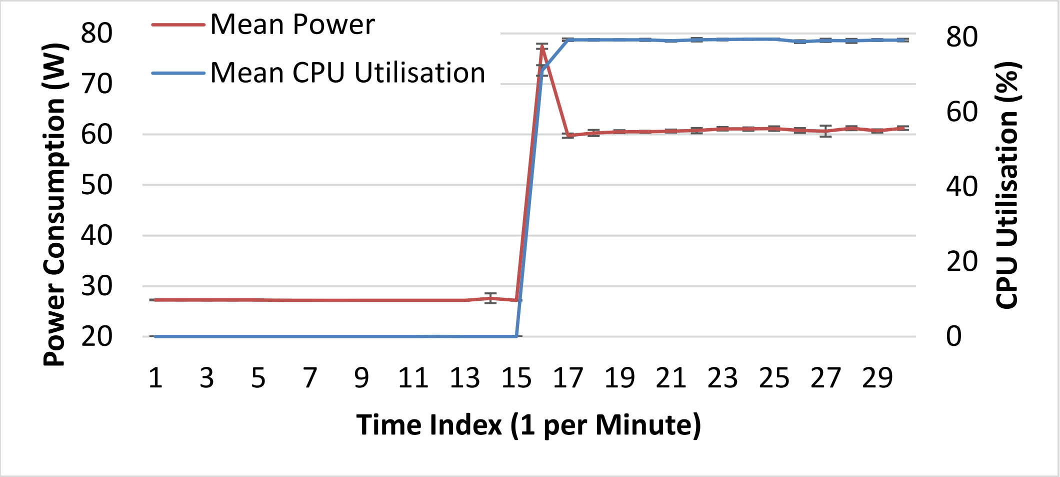

Figure 5-19: Mean Power Consumption and CPU Utilisation for VM_C

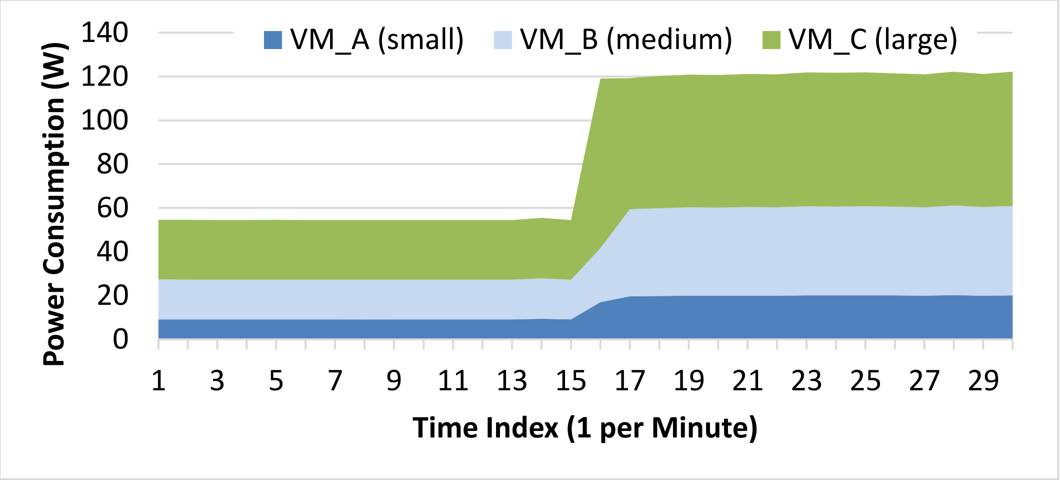

Figure 5-20: PM Mean Power Consumption Attributed to each VM

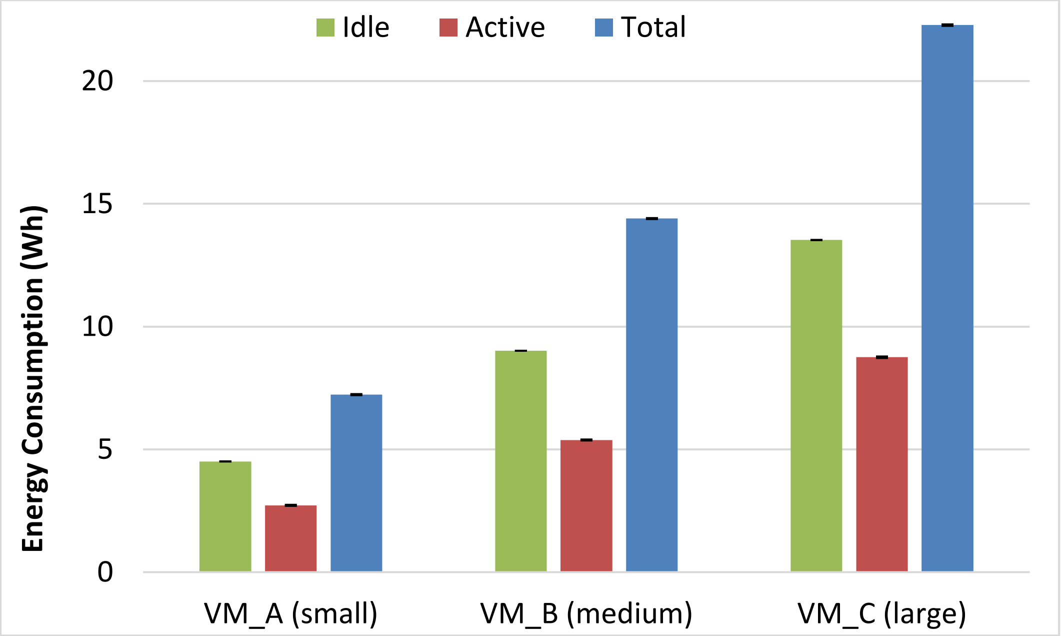

Figure 5-21: Mean Energy Consumption per VM (for 30 minutes)

Figure 5-20 shows the distribution of the PM’s mean power consumption to all these three VMs over time, and Figure 5-21 shows the mean energy consumption per VM in terms of their idle, active and total energy. Like in the previous experiment, the VMs are heterogeneous in terms of the size and therefore have different attribution of the idle and active energy consumption. The energy consumption of VM_A is about two times smaller than VM_B and three times smaller than VM_C, which is fairly based on their CPU utilisation and sizes defined by the number of VCPUs each VM has.

5.4.4.2 Host B

The mean power consumption and CPU utilisation for VM_A, VM_B and VM_C running on Host_B are shown in Figures 5-22, 5-23 and 5-24, respectively. Recall, all of the VMs are idling in the first 15 minutes and actively running with 80% of CPU utilisation for the remaining 15 minutes.

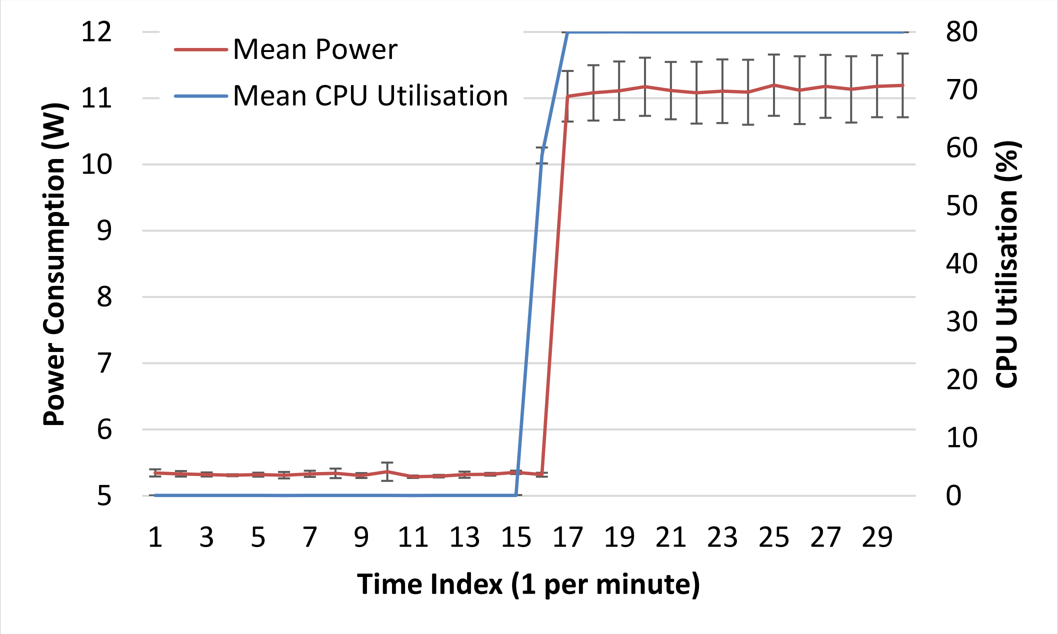

Figure 5-22: Mean Power Consumption and CPU Utilisation for VM_A

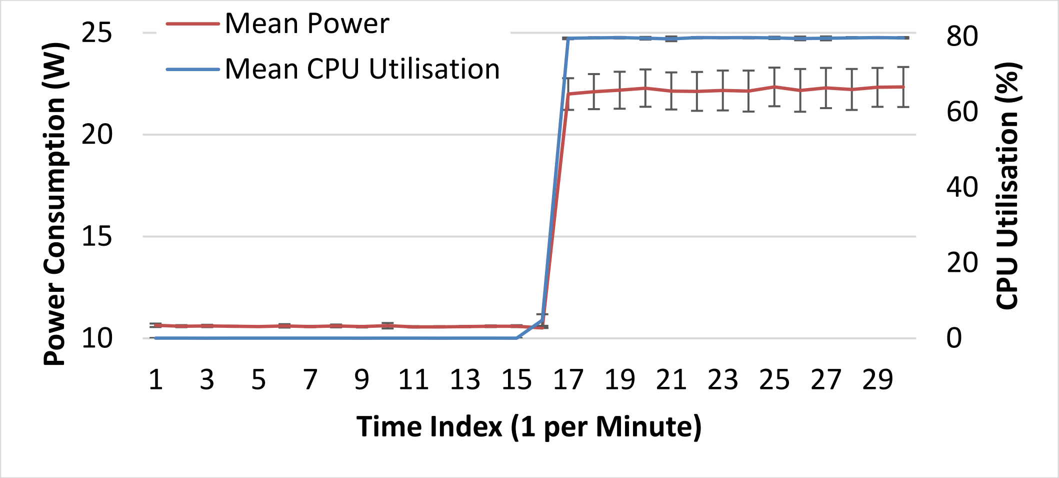

Figure 5-23: Mean Power Consumption and CPU Utilisation for VM_B

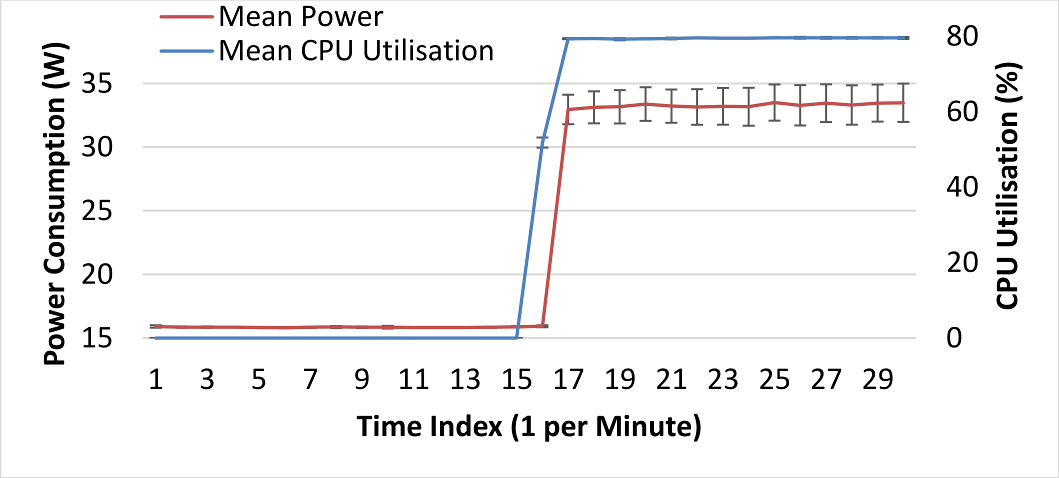

Figure 5-24: Mean Power Consumption and CPU Utilisation for VM_C

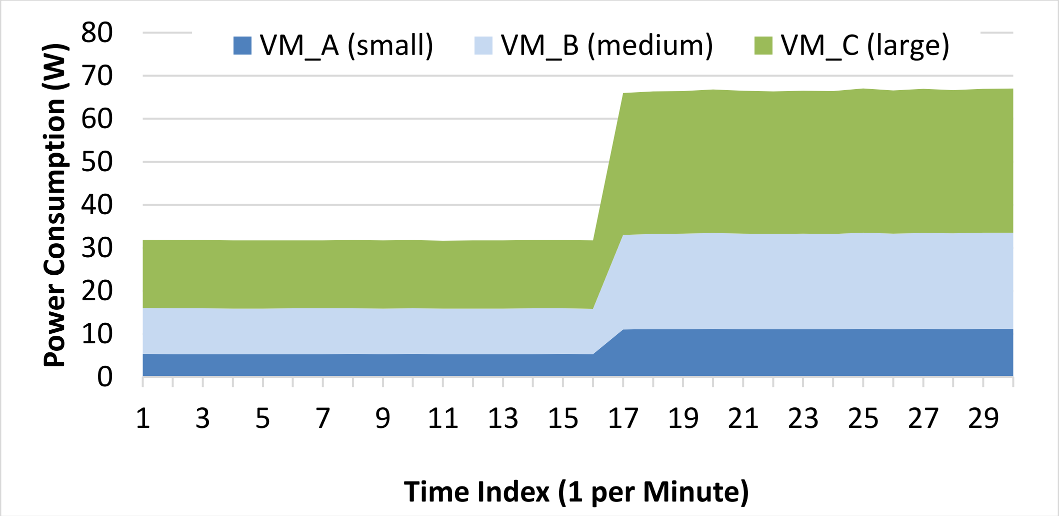

Figure 5-25: PM Mean Power Consumption Attributed to each VM

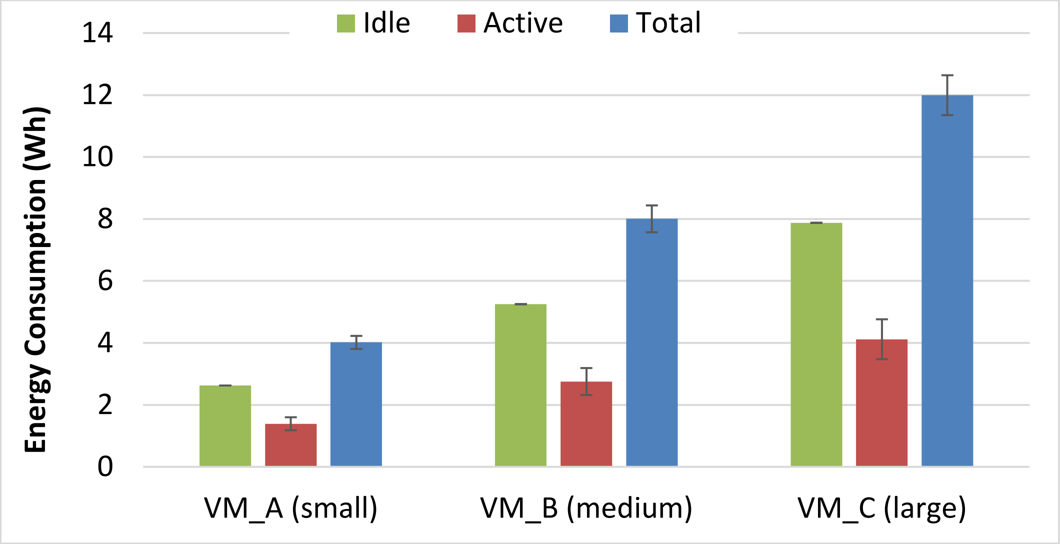

Figure 5-26: Mean Energy Consumption per VM (for 30 minutes)

Figure 5-25 shows the distribution of the PM's mean power consumption to all three VMs, and Figure 5-26 shows the mean energy consumption per VM in terms of their idle, active and total energy. As the VMs are heterogeneous in terms of the size, they consequently have different attribution of the idle and active energy consumption. The energy consumption of VM_A is about two times smaller than VM_B and three times smaller than VM_C, which is fairly based on their CPU utilisation and sizes defined by the number of VCPUs each VM has.

5.4.5 Overall Results Discussion

The energy-aware profiling model has been introduced to fairly attribute the PM's energy consumption to homogeneous and heterogeneous VMs. Based on the results of the conducted experiments, the proposed energy-aware profiling model is capable of fairly attributing the PM's energy consumption to homogeneous VMs, as shown in Section 5.4.2, and to heterogeneous VMs as shown in Section 5.4.3. The attribution mechanism of the PM power consumption considered in this model is based on the VMs' CPU utilisation and their sizes in terms of the number of the VCPUs, which reflects the real usage of the physical

CPU resource impacting the real power usage. Hence, the real power consumption of a PM can be fairly attributed to the VMs owing to the introduced energy-aware profiling model, which addresses the second research question (**Q.2** - see Section 1.3).

The third experiment presented in Section 5.4.4 has shown the energy consumption attribution for three heterogeneous VMs running on the same PM. Also, it has revealed that when these three types of VMs run on a different PM, they can have different attribution of energy consumption based on the power characteristics of the PM. Host_B has less idle and active power consumption than Host_A; therefore, when these three types of VMs are running on Host_A, they have more energy consumption as compared to when running on Host_B, as shown in Figures 5-21 and 5-26, respectively. Hence, enabling energy-awareness at the VM level can help the Cloud service providers to monitor the energy consumption of the VMs and, if necessary, migrate the VMs to another host to maintain their energy goals.

Further, the conducted experiments reveal that a considerably large portion of the VMs' total energy resides on their the idle energy, which is being attributed from the idle energy of the underlying PM. Thus, attributing the PM's idle energy to the VMs, which is already considered in the proposed model, is very important, especially to alleviate the idle energy costs for the PMs, as it has been also argued in [138].

5.5 Summary

The energy-aware profiling model for enabling energy-awareness at the VM level during the operation of Cloud services has been presented and discussed thoroughly. This chapter has presented some experiments conducted on the Cloud testbed to verify that the number of VCPUs is the key parameter reflecting the actual usage of the physical CPU resources, and therefore this parameter has been used in the proposed model. Finally, the energy-aware profiling model has been evaluated in terms of its ability to fairly attribute the PM’s energy consumption to homogeneous and heterogeneous VMs by a number of direct experiments conducted on the Cloud testbed.

The following chapter will discuss the energy-aware prediction framework proposed in this thesis for forecasting the power consumption of the VMs prior to the deployment of Cloud services.

Chapter 6 Energy Prediction

6.1 Overview

This chapter firstly presents the energy-aware prediction framework that aims to forecast the power usage for VMs prior to service deployment, as presented in Section 6.2. This framework works by predicting the VMs workload based on historical workload data that has reoccurring patterns and correlating the predicted VM workload with physical resources to predict the power usage of VMs. A number of experiments along with their results are then presented in Section 6.4 to evaluate this framework for predicting the power consumption of VMs in future run-time.

6.2 Energy-aware Prediction Unit

The previous chapter has presented the EPU unit as the key component of the introduced energy-aware Cloud system architecture for enabling energy-awareness of VMs during the service run-time at the operational level. The key component of this architecture to enable energy-awareness of VMs at future run-time of the services is the EPREU unit, as highlighted in blue in Figure 6-1. The EPREU unit has a framework consisting of a number of models with the overall objective to forecast the energy consumption of VMs prior to service deployment by considering the type of VMs and their historical workload data. The predicted energy consumption of VMs can help service providers to deploy their services with enhanced energy-efficient decisions proactively. The details of the framework are introduced next in Section 6.2.1.

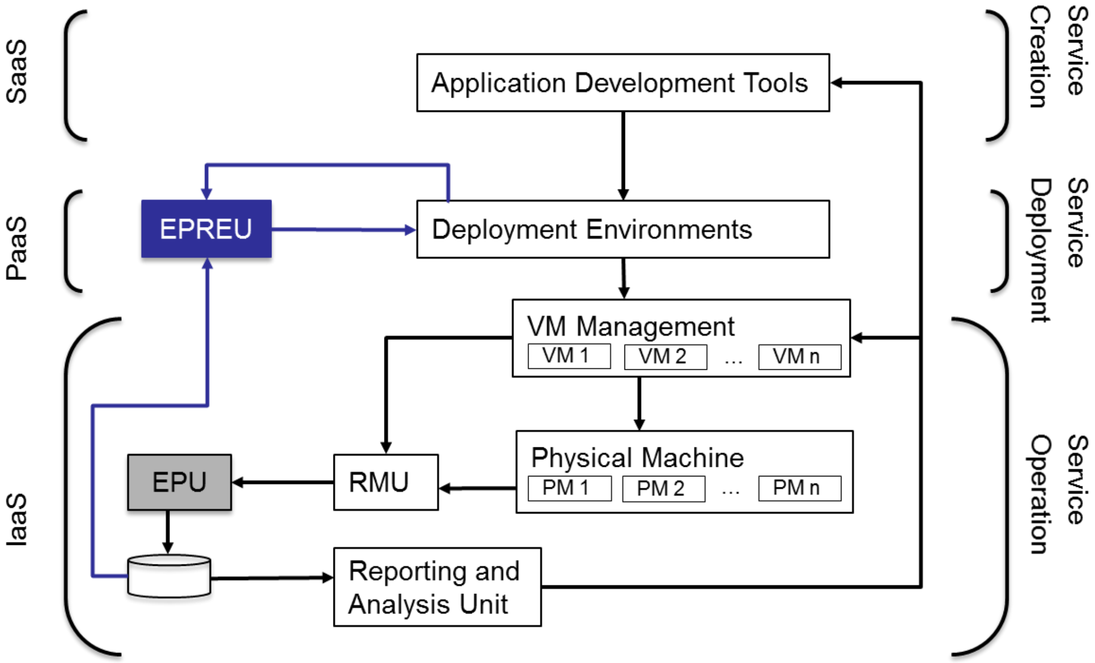

Figure 6-1: Energy-aware Cloud System Architecture - EPREU

6.2.1 Energy-aware Prediction Framework

As measuring the current power consumption is difficult and cannot be performed directly at the VM level, predicting the future power consumption is even more difficult at this level because it would rely on the predicted PM's power to be used. Therefore, an energy-aware prediction framework that aims to forecast the power consumption for the new VMs prior to service deployment is introduced. This framework includes a model that first predicts the workload at the VM level. After that, this predicted VM workload is correlated to PM workload in order to estimate the new PM power consumption, from which the predicted VM power consumption would be based on. As depicted in Figure 6-2, this energy-aware prediction framework includes four main steps in order to forecast the VMs' power consumption.

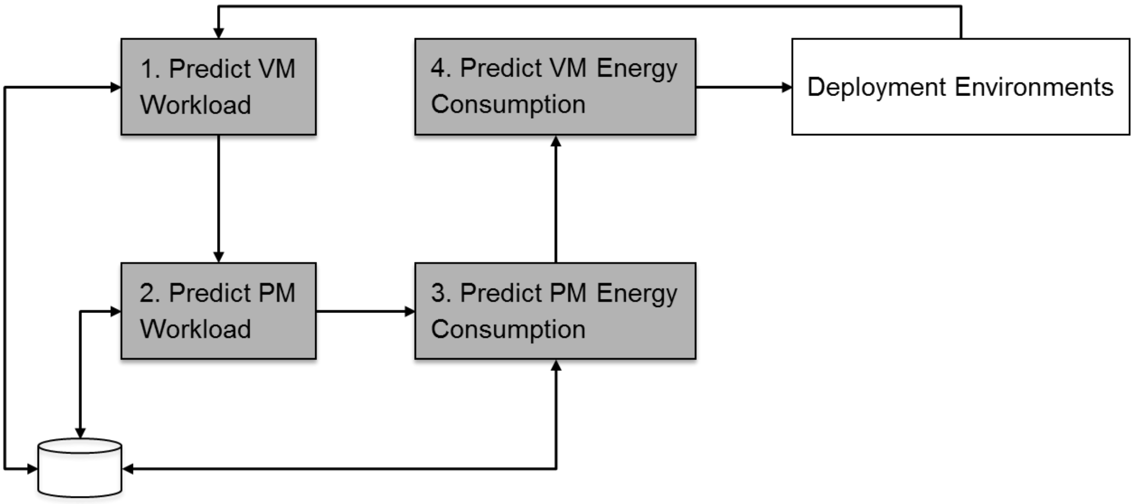

Figure 6-2: Energy-aware Prediction Framework

6.2.1.1 Predict VM Workload

The first step of the framework is to predict the VM workload, which is the VM CPU utilisation. The deployment environment specifies the prerequisite information, which is the requested number of VMs along with their capacity in terms of VCPUs to execute the application, before such deployment process takes place. Using the ARIMA model, the VM workload is then predicted based on historical workload patterns retrieved from a knowledge database. As previously discussed in Section 2.3.3, there are five different types of workload patterns that can be experienced in Cloud applications; and two types of these workload patterns, namely static and periodic, are considered for the historical data to be used in this framework. Thus, this work is limited to only these two workload patterns due to time constraint, and the other patterns are considered as part of the future work. Despite considering only two patterns, this framework presents promising work as being the first for predicting the VM workload driven through Cloud application workload patterns.

The ARIMA model is a time series prediction model that has been used widely in different domains owing to its sophistication and accuracy [162]. As previously discussed, a number of work, as in [151]–[154], have used ARIMA model to predict workload in the Cloud Computing domain; though their objectives do not consider predicting the energy consumption. Hence, the same approach using ARIMA model is applied in this thesis to predict the workload, but with the objective towards predicting the energy consumption of VMs. Unlike other prediction methods, like sample average, ARIMA takes multiple inputs as historical observations and outputs multiple future observations depicting the seasonal trend. It can be used for seasonal or non-seasonal time-series data. The type of seasonal ARIMA model is used in this research as the targeted workload patterns are reoccurring and showing seasonality in time intervals. In order to use the ARIMA model for predicting the VM workload, the historical time series workload data has to be stationary, otherwise it should be transformed to stationary. There are some ready-to-use methods available in any statistical packages, like R package [163], for transforming the non-stationary data to stationary. Some of these methods include data differencing and Box and Cox transformation [164], which are used in this research. Further, the model selection of ARIMA can be automatically processed in R package [163] and the best fit ARIMA model is selected based on Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) value [162].

6.2.1.2 Predict PM Workload

Once the VM’s workload is predicted, the second step is to understand how this workload would be reflected on the physical resources and predict the new PM’s workload, which is PM CPU utilisation, with consideration of its current workload as the PM may be running other VMs. Therefore, the relationship between the number of VCPUs and the PM's CPU utilisation should be characterised for the targeted PMs. For the purpose of this research, two different PMs in a Cloud testbed have been characterised with regression models, as presented in Section 6.3.1. Thus, based on the linear relation equation of VCPUs and CPU utilisation for each PM, the new increment of PM's CPU utilisation can be estimated by considering the used ratio of the requested VCPUs for the VMs, $VM_{xReqVCPUs}$, identified by the predicted VM CPU utilisation, $VM_{xPredUtil}$. VM_{Count} is the number of VMs requested to run on the same PM. This new increment of PM's CPU utilisation would need to be added to the current PM's CPU utilisation, $PM_{xCurrUtil}$, in order to identify the new total of the predicted PM's CPU utilisation, $PM_{xPredUtil}$, as described in Equation 6.1. Alpha, α, and beta, β, are the interceptor and slope values obtained in the linear regression relation of VCPUs and CPU utilisation for each PM. The PM's idle CPU utilisation, $PM_{xIdleUtil}$, is subtracted from the current PM's CPU utilisation because the relation equation already considers this idle value.

$$PM_{xPredUtil} = \left(\beta \times \left(\sum_{y=1}^{VM_{Count}} (VM_{xReqVCPUs} \times \frac{VM_{xPredUtil}}{100})\right) + \alpha\right) + (PM_{xCurrUtil} - PMx_{IdleUtil}) \quad (6.1)$$

6.2.1.3 Predict PM Energy Consumption

After predicting the PM's workload, the third step is to predict the PM's energy consumption based on the correlation of this predicted workload with PM energy consumption. Thus, the considered PMs in the Cloud testbed need to be characterised in terms of their power consumption relation with CPU utilisation using regression models, as presented in Section 6.3.1.

Hence, the PM's predicted power consumption, $PM_{xPredPwr}$, can be identified using a linear relation with the predicted PM's CPU utilisation, as shown in Equation 6.2. α and β are the interceptor and slope values obtained from the regression relation.

$$PM_{xPredPwr} = \beta \times PM_{xPredUtil} + \alpha \tag{6.2}$$

However, as discussed previously, not all existing PMs may necessarily follow a linear power model with their CPU utilisation. In this case, other regression models, e.g. polynomial, can be investigated and used to identify the relation between the power consumption and CPU utilisation of the targeted PM.

6.2.1.4 Predict VM Energy Consumption

The final step within this framework is to profile and attribute the predicted PM's energy consumption to the new requested VM and to the VMs already running on that physical host based on the energy-aware profiling model introduced in Section 5.2.1. Thus, the predicted power consumption for the new VM, $VM_{xPredPwr}$, prior to deployment can be identified for the next interval time using Equation 6.3.

$$\begin{aligned} VM_{xPredPwr} = {} & PM_{xIdlePwr} \times \frac{VM_{xReqVCPUs}}{\sum_{y=1}^{VM_{Count}} VM_{yVCPU}} \\ & + (PM_{xPredPwr} - PM_{xIdlePwr}) \\ & \times \frac{VM_{xPredUtil} \times VM_{xReqVCPU}}{\sum_{y=1}^{VM_{Count}} (VM_{yUtil} \times VM_{yVCPU})} \end{aligned} \tag{6.3}$$

6.3 Implementation

The energy-aware prediction framework is introduced in this research to predict the power consumption of VMs prior to service deployment based on historical static and periodic workload patterns. Thus, in order to evaluate this framework, a number of direct experiments have been conducted on the Leeds Cloud testbed to synthetically generate historical workload data. The historical data has been generated to represent real workload patterns of Cloud applications (discussed in Section 2.3.3), including static and periodic, by stressing the CPU on different types of VMs with the Stress tool [160] (see Section 5.4.1). The type of VMs is identified by their size in terms of the number of VCPUs, e.g. a small VM with one VCPU and a large VM with three VCPUs. The generated workload of each VM type has four time intervals of 30 minutes each. The first three intervals will be used as the historical data set for prediction, and the last interval will be used as the testing data set to evaluate the predicted results.

The prediction process works offline by firstly predicting the VM workload using the ***auto.arima*** function in R package [163] to automatically select the best fit model of ARIMA based on AIC or BIC value. Once the VM workload is predicted, the process is then completed by going through the steps of the introduced framework to consider the correlation between the physical and virtual resources and consequently predict the power consumption of VMs prior to their deployment on PMs. As a key requirement of the framework, the targeted PMs should be characterised for once in terms of the relation between the number of VCPUs and the CPU utilisation as well as the relation between the CPU utilisation and power consumption. Therefore, two different PMs in the Cloud testbed have been characterised, as presented next in Section 6.3.1.

6.3.1 Characterisation of Physical Machines

The CPU usage of a VM has an effect on the underlying PM’s CPU usage, which also impacts the PM’s power consumption. As a VM can have and use one or many VCPUs, it is important to understand the correlation between the targeted PM's CPU utilisation and the number of used VCPUs. In the context of the introduced framework, it is also important to understand the correlation between the CPU utilisation and power consumption of a PM.

In this regard, two different PMs on the Leeds Cloud testbed have been considered. The first PM, Host_A, has a four core X3430 Intel Xeon CPU, and the second PM, Host_B, has a eight core E3-1230 V2 Intel Xeon CPU. Two direct experiments have been conducted with the aim to 1) understand the relation between each PM's CPU utilisation and the number of VCPUs, and 2) understand the power characteristics of each PM with their CPU utilisation.

6.3.1.1 Host A

The first experiment is designed to run a VM having four VCPUs at five stages on Host_A, with each stage running for five minutes. The VM is idling at the first stage and not utilising any VCPUs, and for the subsequent stages the VM is scaling up utilising one to four VCPUs.

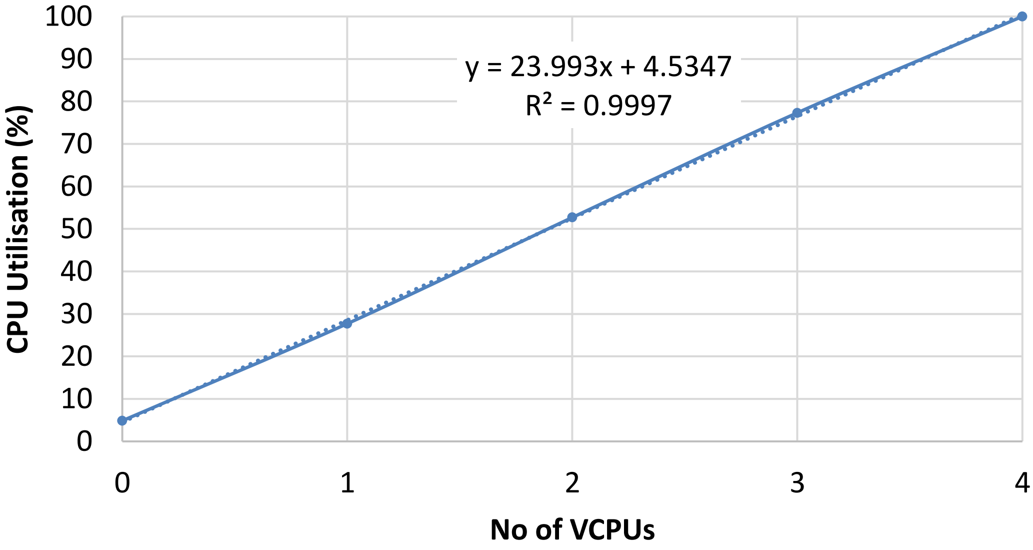

Figure 6-3: Number of VCPUs vs CPU Utilisation for Host_A

By averaging the results of each stage, Figure 6-3 reveals a linear relation between the number of VCPUs and CPU utilisation for Host_A. Also, Figure 6-4 reveals a linear relation between the CPU utilisation and power consumption for Host_A.

The standard deviations of the CPU utilisation and power consumption for each of the five stages in this experiment are shown in Table 6-1, as they are very small and not noticeable in the figures.

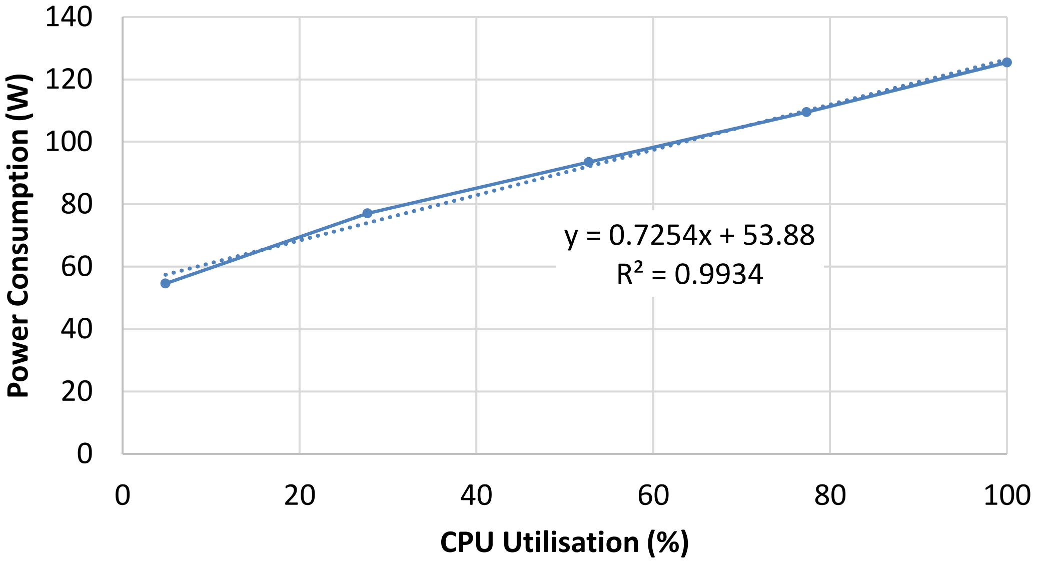

Figure 6-4: CPU Utilisation vs Power Consumption for Host_A

Table 6-1: Standard Deviation of the CPU Utilisation and Power Consumption for each Stage

Stage	CPU Utilisation	Power Consumption
1	0.12	0.09
2	0.04	0.19
3	0.03	0.24
4	0.05	0.21
5	0	0.2

Thus, using the linear relation along with the identified values of the slope and interceptor, as shown on Figure 6-3, can help predict the CPU utilisation of Host_A based on the ratio of the used VCPUs by the VMs, as discussed in Section 6.2.1.2. Likewise, using the linear relation shown on Figure 6-4 can help predict the power consumption of Host_A based on its predicted CPU utilisation, as discussed in Section 6.2.1.3.

6.3.1.2 Host B

Similarly, the second experiment is designed to run a VM having eight VCPUs at nine stages on Host_B, with each stage running for five minutes. The VM is idling at the first stage and not utilising any VCPUs, and for the subsequent stages the VM is scaling up utilising one to eight VCPUs.

By averaging the results of each stage, Figure 6-5 reveals a linear relation between the number of VCPUs and CPU utilisation for Host_B. Yet, Figure 6-6 reveals when the CPU utilisation is in the range 50% - 100%, the power consumption almost stabilises and does not increase further. With the linear model shown in Figure 6-6, the power consumption of Host_B is overestimated when the CPU utilisation is less than 15% and above 80%, and underestimated when the CPU utilisation is between 15% and 80%.

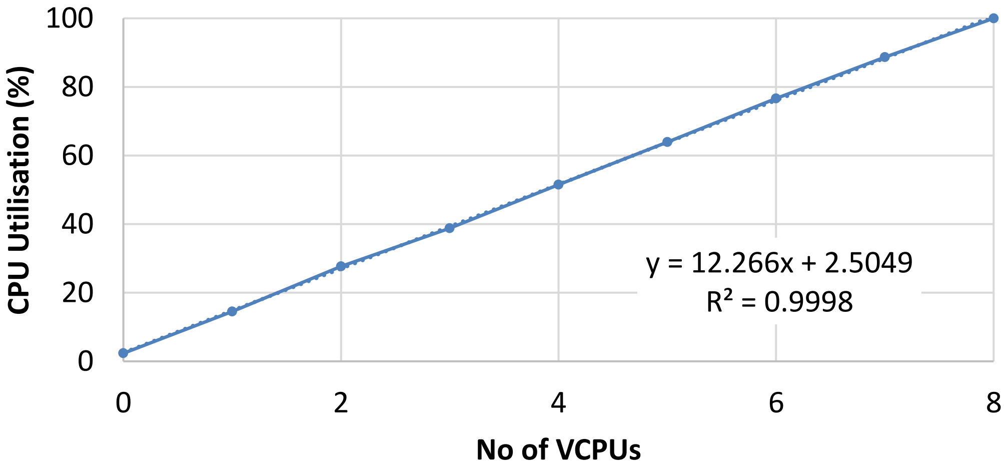

Figure 6-5: Number of VCPUs vs CPU Utilisation for Host_B

Therefore, a linear power model is not suitable for Host_B, and another regression model should be investigated to describe the power relation with the CPU utilisation. As shown in Figure 6-7, the relation of CPU utilisation and power consumption for Host_B can be suitably described using a polynomial model with order three.

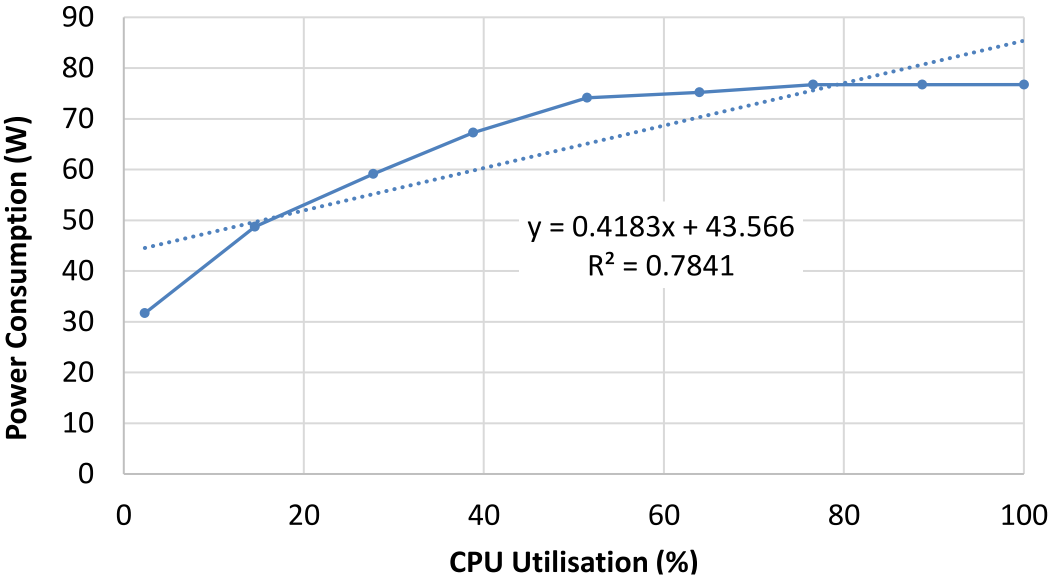

Figure 6-6: CPU Utilisation vs Power Consumption for Host_B

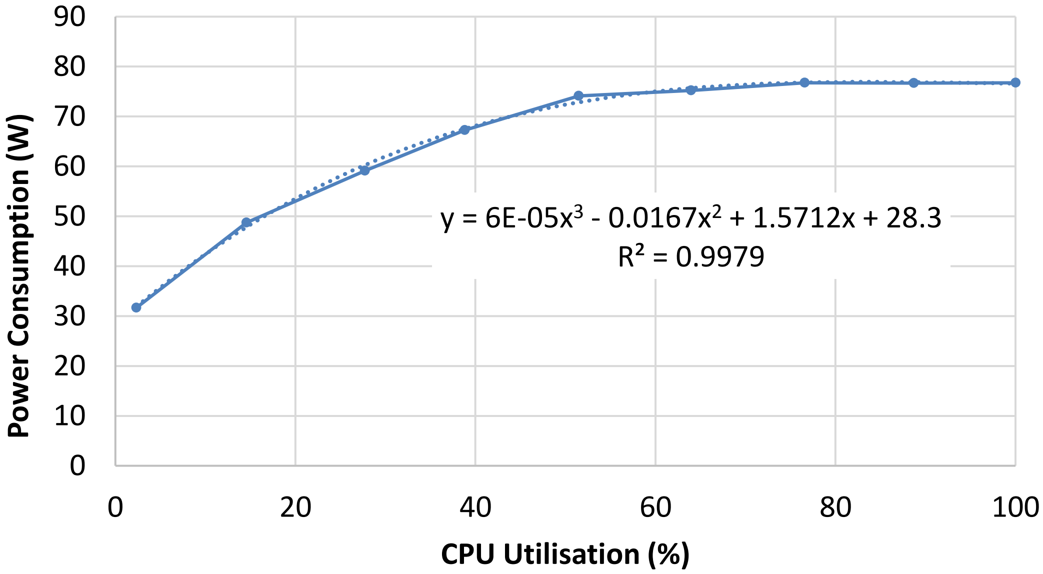

Figure 6-7: CPU Utilisation vs Power Consumption for Host_B

The standard deviations of the CPU utilisation and power consumption for each of the nine stages in this experiment are shown in Table 6-2.

Table 6-2: Standard Deviation of the CPU Utilisation and Power Consumption for each Stage

Stage	CPU Utilisation	Power Consumption
1	0.08	0.08
2	1.09	0.07
3	2.47	0.15
4	0.02	0.11
5	0.03	0.27
6	0.03	0.2
7	0.46	0.33
8	0.05	0.22
9	0	0.18

All in all, using the linear relation along with the identified values of the slope and interceptor, as shown on Figure 6-5, can help predict the CPU utilisation of Host_B based on the ratio of the used VCPUs by the VMs, as discussed in Section 6.2.1.2. Likewise, the polynomial relation shown on Figure 6-7 can help predict the power consumption of Host_B based on its predicted CPU utilisation, as discussed in Section 6.2.1.3.

6.4 Experiments and Evaluation

6.4.1 Design of Experiments

The overall aim of the experiments is evaluate the energy-aware prediction framework for forecasting the power consumption of the VMs prior to service deployment based on historical static and periodic workload patterns. As discussed in Section 6.3, historical static and periodic workload patterns have been generated synthetically by conducting a number of experiments stressing the CPU utilisation on different types of VMs. In order to generate a static workload pattern, each VM type is run at 80% of CPU utilisation for four repeated time intervals, with 30 minutes runtime in each interval. To generate a periodic workload pattern, each VM type has four repeated time intervals, with each interval running for 30 minutes and having two peaks of CPU utilisation at 80%. The reason of having four time intervals is to use the first three intervals as historical data set for prediction and the last interval as the testing data set for evaluation purposes. A similar approach is used in [154] and followed in this research.

The generated historical data, both VM workload and their power consumption, will be presented first in each experiment. The first three intervals of the generated VM workload are only used as historical data set by the framework to predict VM workload and power consumption for the next time interval. The last interval of the generated VM workload and power consumption will be used as testing data set to evaluate the predicted results.

In terms of the design of the experiments:

1) The first experiment is designed to forecast the workload and power consumption of a large VM based on historical static workload pattern, as presented in Section 6.4.2.
2) The second experiment is designed to forecast the workload and power consumption of a large VM based on historical periodic workload pattern, as presented in Section 6.4.3.

3) The third experiment is designed to forecast the workload and power consumption of two types of VMs, small and large, each one based on different historical workload patterns, static and periodic, respectively. Also, this third experiment is designed to forecast the workload and power consumption of these two VMs when being run on two different PMs with different characteristics, as presented in Section 6.4.4. Thus, the aim of the third experiment is to evaluate the prediction for heterogeneous types of VMs with a mix of workload patterns when being run on two different hosts, having different power characteristics.

For all experiments, the mean value along with the higher and lower values of 95 and 80 percent confidence intervals are considered and shown for the predicted workload of each VM based on the ARIMA model. Also, these predicted VM workload values are correlated with the physical resources via the proposed framework to get the mean along with the higher and lower values of the 95 and 80 percent confidence intervals for the predicted power consumption of each VM.

6.4.2 Experiment 1: Large VM with Static Workload Pattern on a Single Host

This experiment shows the prediction results for a large VM about to run in a single PM, Host_A, based on historical static workload pattern. In terms of the historical and testing data sets used in this experiment, Figure 6-8 presents the generated VM workload along with its power consumption in four time intervals.

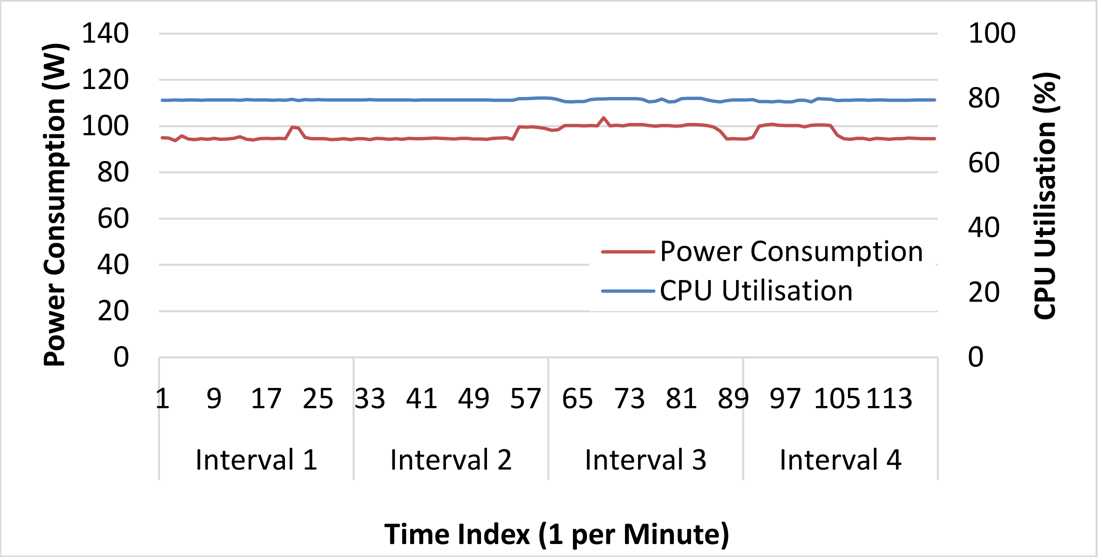

Figure 6-8: Historical Data for a Large VM Based on Static Workload

As discussed previously, the process of VM workload prediction within the framework uses the ARIMA model to forecast the next 30 minute period of workload, as shown in Figure 6-9, based on three historical intervals of workload data. Overall, the predicted VM workload results closely match the actual workload owing to the sophistication of the ARIMA model.

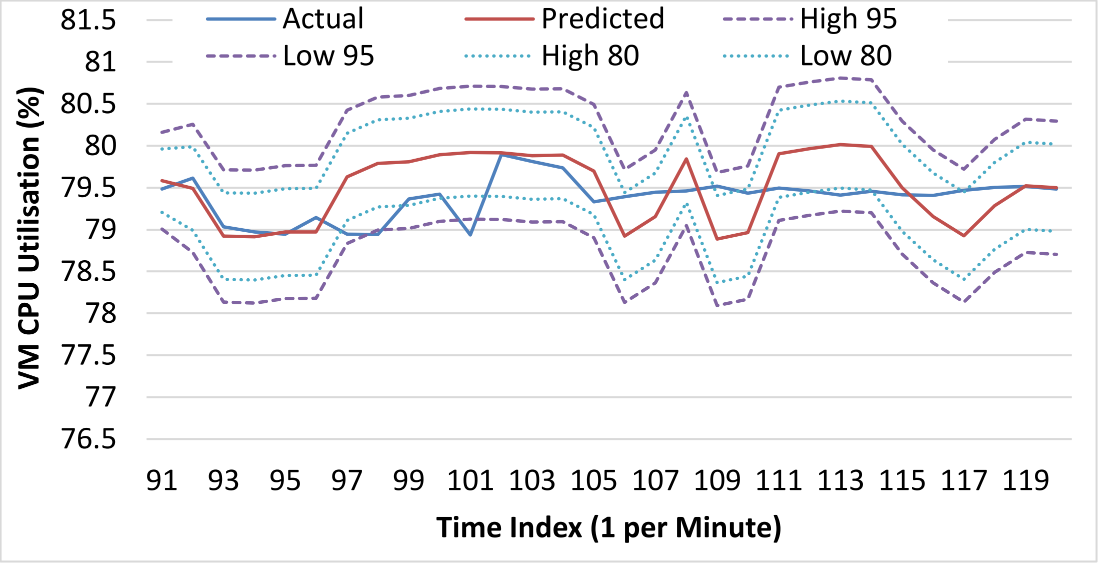

Figure 6-9: Workload Prediction for a Large VM Based on Static Workload Pattern

Based on this predicted workload, the VM power consumption is predicted using the remaining models within the framework, as previously discussed in Section 6.2.1.

Figure 6-10 shows the predicted VM power consumption results, which have a small variation as compared to the actual power consumption. The reason of this variation is because there is an accumulation of errors from the previous steps within the framework, especially when correlating the PM's CPU utilisation to the PM's power consumption.

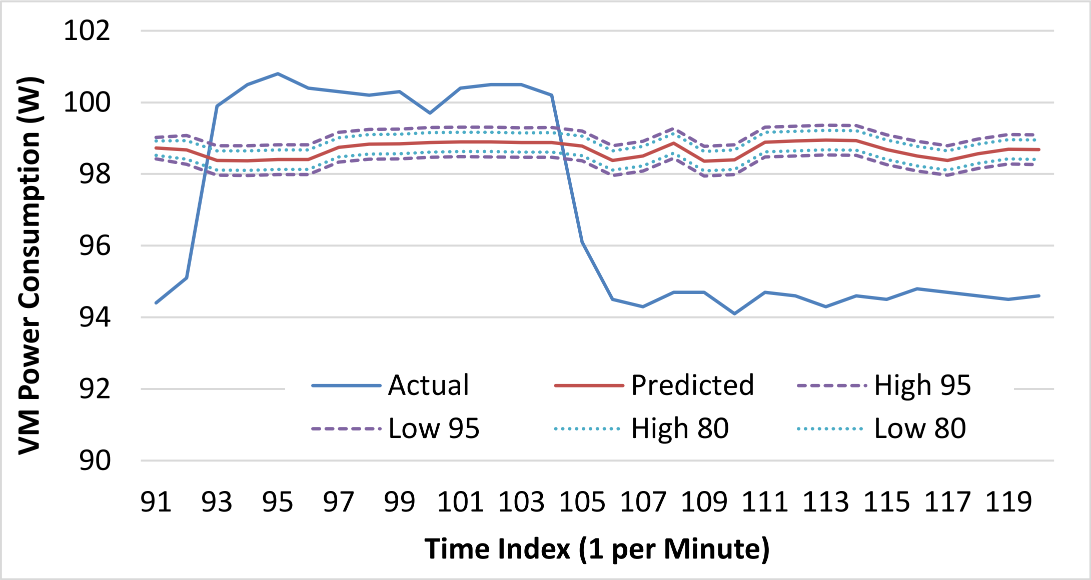

Figure 6-10: Power Prediction for a Large VM Based on Static Workload Pattern

As shown on Figure 6-10, the actual power consumption increases in the first part of the interval; this may be due to the thermal energy, which is not captured in this work, causing the machine's fan to run faster and thus leading to an increase of PM power, which is then attributed to the VM. Despite this accumulation of error, the proposed framework can predict the VM power consumption accurately.

In terms of prediction accuracy, a number of metrics described in [165], [166] are used to evaluate the predicted VM workload and power consumption based on static workload, as shown in Table 6-3. The error value is calculated as the difference between the observed (actual) value and the predicted value, and the mean Error (ME) is used to calculate the average of all errors within the data set; the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are scale-dependant accuracy measures that can be used when comparing between data sets having the same scale; the Mean Percentage Error (MPE) and Mean Absolute Percentage Error (MAPE) are scale-independent accuracy measures that can be used when comparing between different data sets [165], [166].

As previously discussed in Section 6.4.1, the actual data of the VM workload and power consumption is used as the testing data set for evaluation purposes.

Table 6-3: Prediction Accuracy for a Large VM Based on Static Workload Pattern

Accuracy Metric	Predicted VM Workload	Predicted VM Power Consumption
ME	-0.11	-1.75
RMSE	0.42	3.28
MAE	0.33	3.04
MPE	-0.14	-1.89
MAPE	0.42	3.17

The accuracy of the predicted VM workload is very high as its metrics' values are close to zero. The predicted VM power consumption is less accurate as compared with the predicted VM workload, but still achieves a good prediction accuracy, with -1.89 of MPE. The reason of the predicted VM power consumption being less accurate than the predicted workload when compared to the actual data is due to the accumulated error when correlating this VM workload to physical resources.

6.4.3 Experiment 2: Large VM with Periodic Workload Pattern on a Single Host

In this experiment, the prediction results for a large VM about to run in a single PM, Host_A, based on historical periodic workload pattern are presented. Figure 6-11 presents the generated VM workload along with its power consumption used in this experiment as the historical and testing data sets.

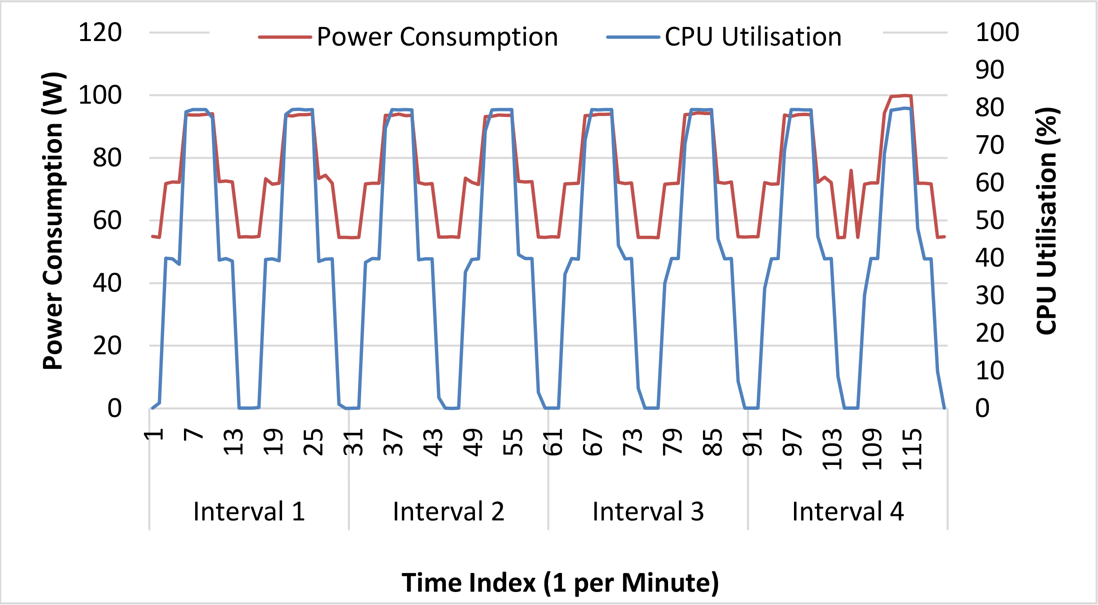

Figure 6-11: Historical Data for a Large VM Based on Periodic Workload

Figure 6-12 shows the results of the predicted versus the actual VM workload. Despite the periodic utilisation peaks, the predicted VM workload results closely match the actual results, which reflects the capability of the ARIMA model to capture the historical seasonal trend and give a very accurate prediction accordingly.

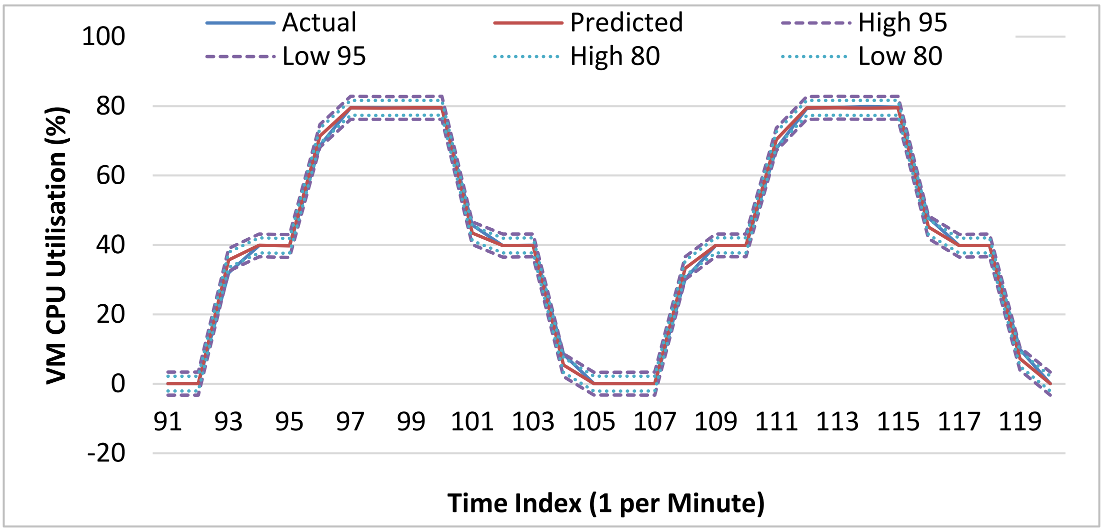

Figure 6-12: Workload Prediction for a Large VM Based on Periodic Workload Pattern

The proposed framework is also capable of predicting the power consumption of the VM with only a small variation as compared to the actual one. As shown in Figure 6-13, the actual VM power consumption in the middle of the interval shows a small peak, which was not followed by the predicted VM power consumption. This is again can be due to the thermal energy which is not considered in this work.

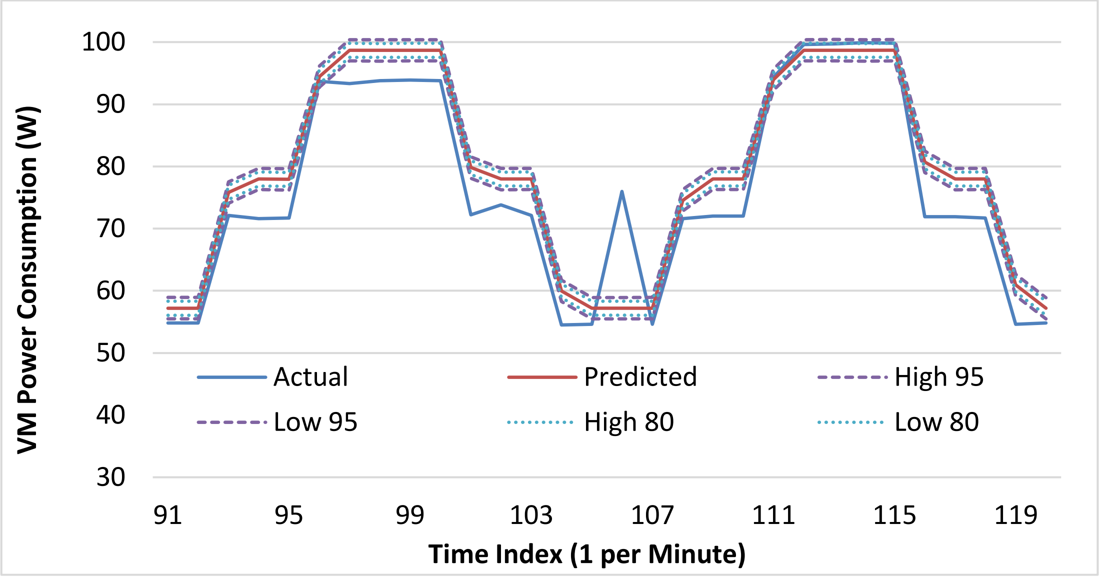

Figure 6-13: Power Prediction for a Large VM Based on Periodic Workload Pattern

For evaluating the accuracy of the predicted VM workload and power consumption based on periodic workload, different accuracy metrics are used, as shown in Table 6-4.

Table 6-4: Prediction Accuracy for a Large VM Based on Periodic Workload Pattern

Accuracy Metric	Predicted VM Workload	Predicted VM Power Consumption
ME	-0.02	-3.04
RMSE	1.51	5.76
MAE	0.81	4.61
MPE	2.58	-4.47
MAPE	5.30	6.43

Despite the high variation of the workload utilisation in the periodic pattern, the accuracy metrics indicate that the predicted VM workload achieves a good accuracy, with 2.58 of MPE. As previously discussed, the accumulated error when correlating the predicted VM workload with the physical resources in order to get the power affects the accuracy of the predicted VM power consumption. Therefore, the predicted VM power consumption is less accurate as compared with the predicted VM workload, but still achieves a good prediction accuracy with -4.47 of MPE.

6.4.4 Experiment 3: Heterogeneous VMs and Workload Patterns on Different Hosts

This experiment shows the prediction results for two types of VMs, a small VM and a large VM, based on static and periodic workload patterns, respectively, when running on two different PMs, having different characteristics. Section 6.4.4.1 presents the prediction results of these two VMs prior to deployment on

Host_A, and Section 6.4.4.2 presents the prediction results of these two VMs to be deployed on Host_B. The aim of this experiment is to evaluate the capability of the proposed framework to predict the power consumption for a mix of VMs with a mix of workload patterns when being run on different PMs.

6.4.4.1 Host A

In terms of the historical and testing data sets, Figure 6-14 shows the generated workload along with the power consumption for the small VM, and Figure 6-15 shows the generated workload along with the power consumption for the large VM, with both VMs running on Host_A at the same time.

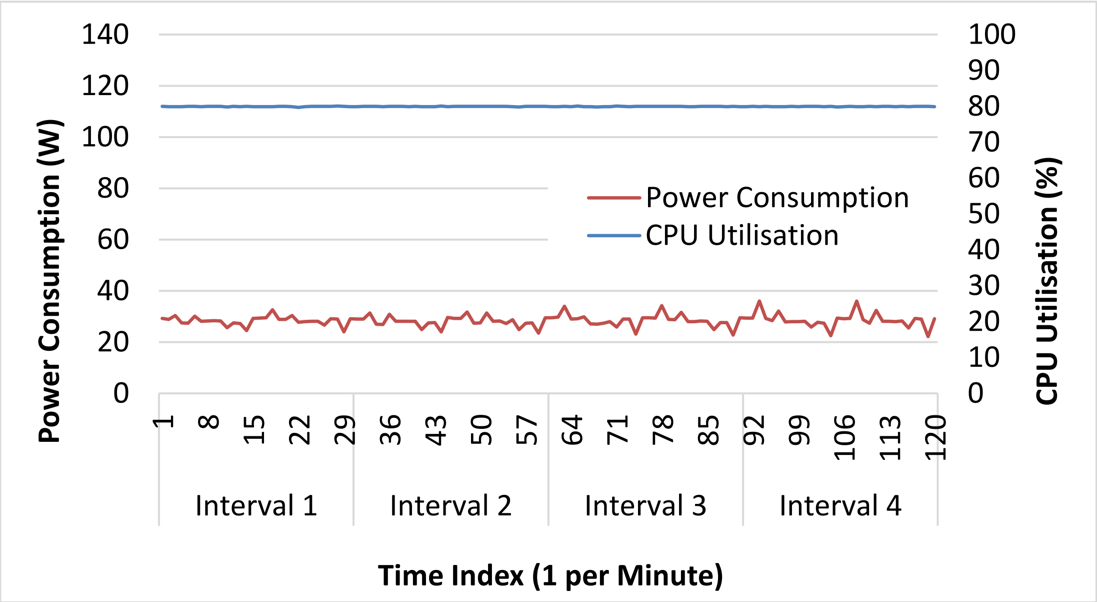

Figure 6-14: Historical Data for a Small VM Based on Static Workload

By using the ARIMA model within the proposed framework to forecast each VM workload, Figure 6-16 shows the predicted results versus the actual for the small VM, and Figure 6-17 shows those results for the large VM.

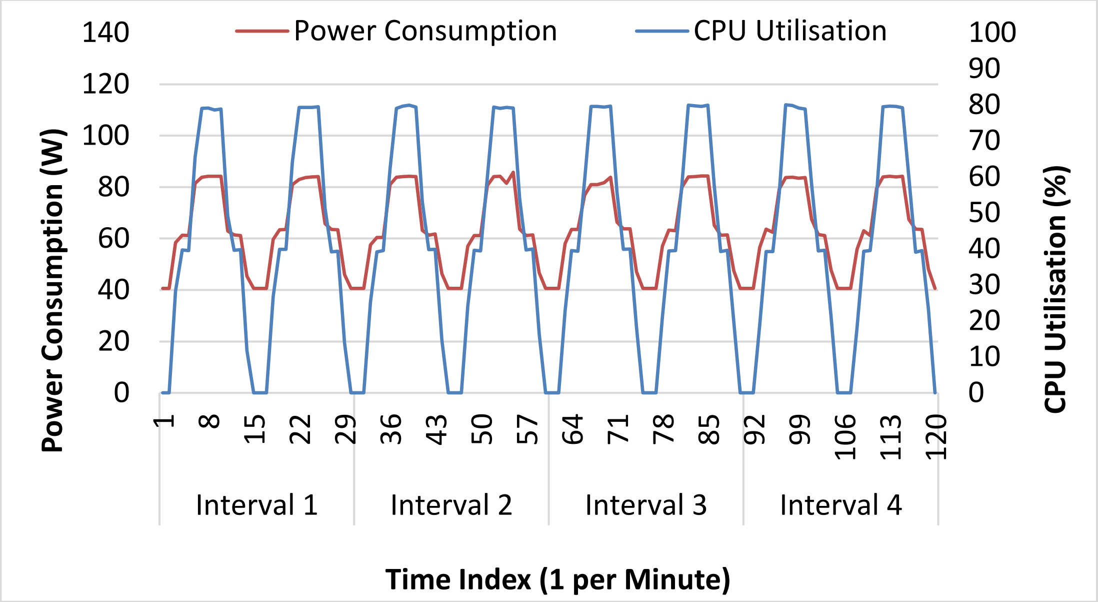

Figure 6-15: Historical Data for a Large VM Based on Periodic Workload

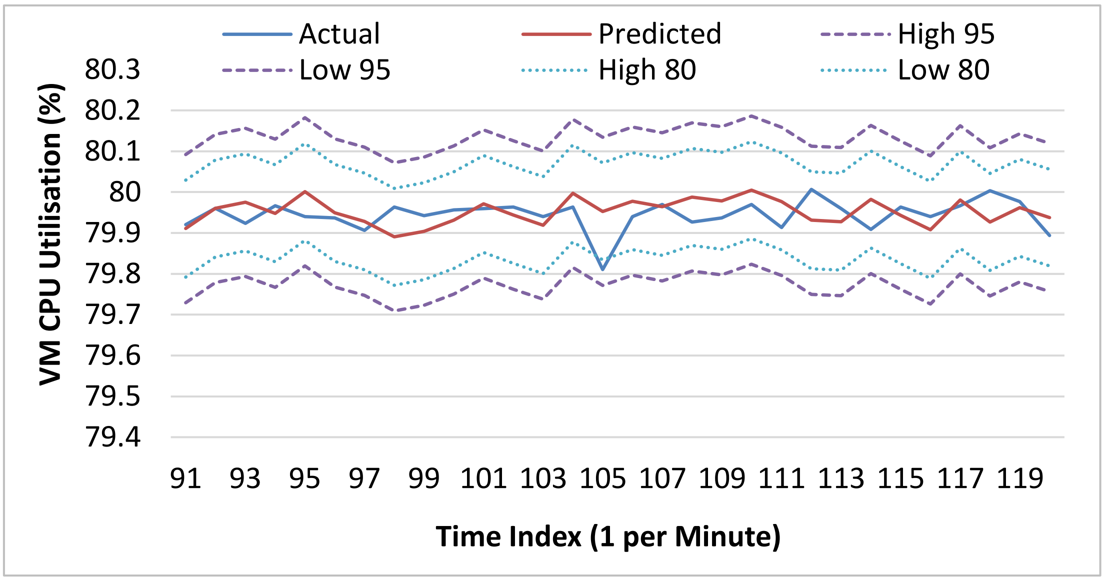

Figure 6-16: Workload Prediction for a Small VM Based on Static Workload Pattern

Overall, the predicted static workload for the small VM closely matches the actual workload, as depicted on Figure 6-16. Also, the predicted periodic workload for the large VM matches the actual workload, as shown on Figure 6-17. Recall, this shows the strength of the ARIMA model for forecasting based on historical seasonal data, repeated patterns of the static and periodic workload.

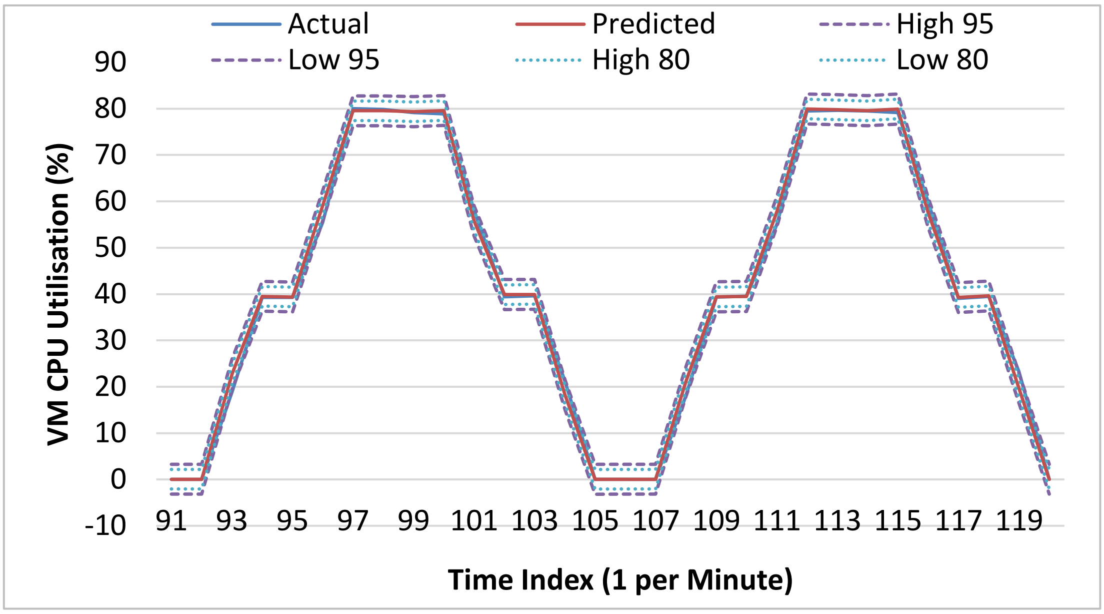

Figure 6-17: Workload Prediction for a Large VM Based on Periodic Workload Pattern

Based on the predicted workload for each VM, their power consumption is predicted via the remaining steps within the framework. Figures 6-18 and 6-19 show the predicted versus the actual results of the power consumption for the small VM and the large VM, respectively.

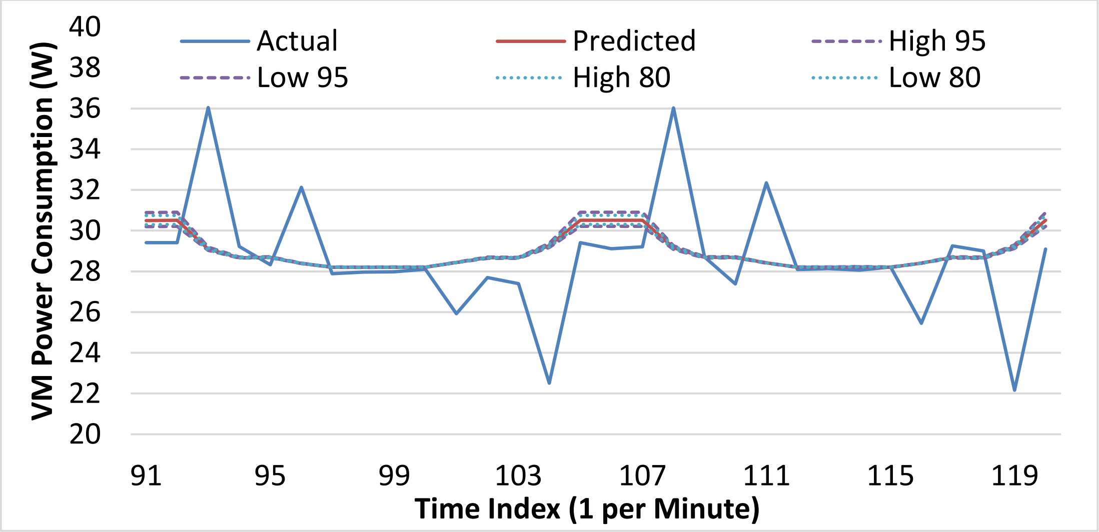

Figure 6-18: Power Prediction for a Small VM Based on Static Workload Pattern

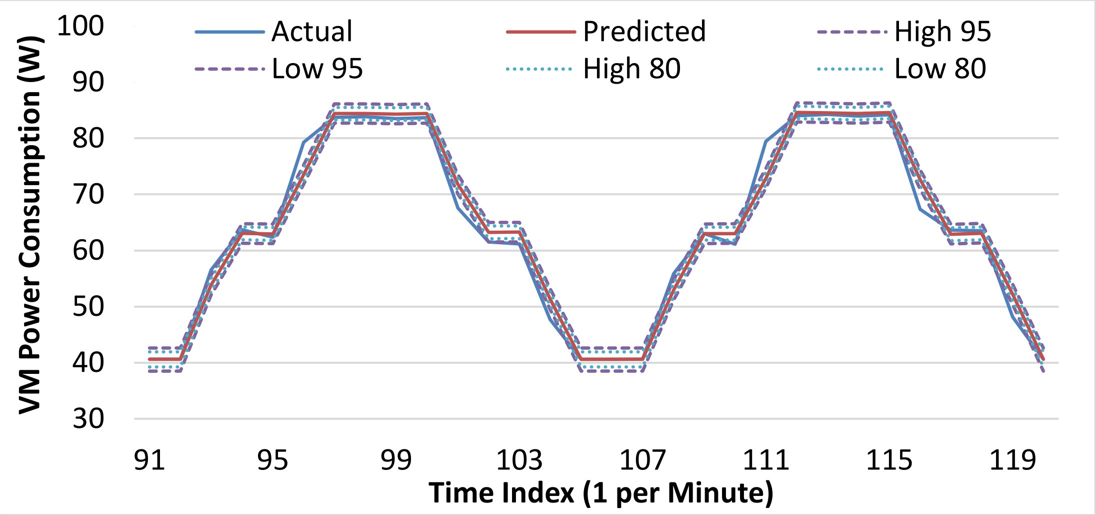

Figure 6-19: Power Prediction for a Large VM Based on Periodic Workload Pattern

The predicted power consumption results for the small VM has a few variations as compared with its actual power consumption. As shown in Figure 6-18, the actual power of the small VM fluctuates while the predicted power almost does not change over the whole interval period as it follows the predicted static workload of this VM; the reason of the changes of the small VM's actual power is because of the changes in the actual PM's power consumption during which the large VM's workload starts to fluctuate for each of its two utilisation peaks within the same interval period. Hence, the predicted power consumption of the large VM, as shown on Figure 6-19, closely matches the actual power consumption since its periodic workload utilisation is the cause of the PM's actual power to vary which follows the periodic pattern of this VM.

In terms of the accuracy metrics, Table 6-5 shows the evaluation of the predicted workload and power consumption for the small VM based on static workload pattern, and Table 6-6 shows the evaluation of the predicted results for the large VM based on periodic workload pattern.

Table 6-5: Prediction Accuracy for a Small VM Based on Static Workload Pattern

Accuracy Metric	Predicted VM Workload	Predicted VM Power Consumption
ME	-0.008	-0.29
RMSE	0.04	2.88
MAE	0.03	1.82
MPE	-0.01	-1.99
MAPE	0.04	6.47

Table 6-6: Prediction Accuracy for a Large VM Based on Periodic Workload Pattern

Accuracy Metric	Predicted VM Workload	Predicted VM Power Consumption
ME	-0.17	-0.27
RMSE	1.61	2.47
MAE	0.96	1.59
MPE	-0.44	-0.58
MAPE	6.56	2.47

Despite the combination of different types of VMs with different workload patterns running on the same PM, the accuracy metrics, shown in Tables 6-5 and 6-6, reveal that the predicted workload of the VMs achieves a good prediction accuracy, with -0.01 of MPE for the small VM and -0.44 for the large VM. Also, the results show a good prediction accuracy in terms of the predicted power consumption for the small VM with -1.99 of MPE and -0.58 for the large VM. Similar to the previous experiments, the accuracy of the predicted VMs power consumption is less that the predicted VMs workload due to the accumulation of error within the process of the proposed framework when correlating the VMs predicted workload with the physical resources to estimate the VMs power prediction. Further, the prediction accuracy of the large VM's workload is less than the small VM's workload because of the high variation of the workload utilisation in the periodic pattern. However, the prediction accuracy of the large VM's power consumption is more accurate than the small VM's power consumption; as explained earlier, when the large VM's actual workload utilisation fluctuates in each of its two peaks within the interval period, it affects the PM's actual CPU resources and therefore the PM's actual active power consumption to fluctuate as well. Hence, this change of PM's active power is attributed to the VMs' actual active power consumption, which matches the periodic pattern of the large VM but not the static pattern of the small VM.

6.4.4.2 Host B

In terms of the historical and testing data sets of the VMs on Host_B, Figure 6-20 shows the generated workload and the power consumption for the small VM, and Figure 6-21 shows the generated workload along with the power consumption for the large VM.

The workload prediction results obtained using the ARIMA model versus the actual values for the small VM and the large VM are shown in Figures 6-22 and 6-23, respectively. As depicted on these two figures, the predicted workload of both VMs matches their actual workload values.

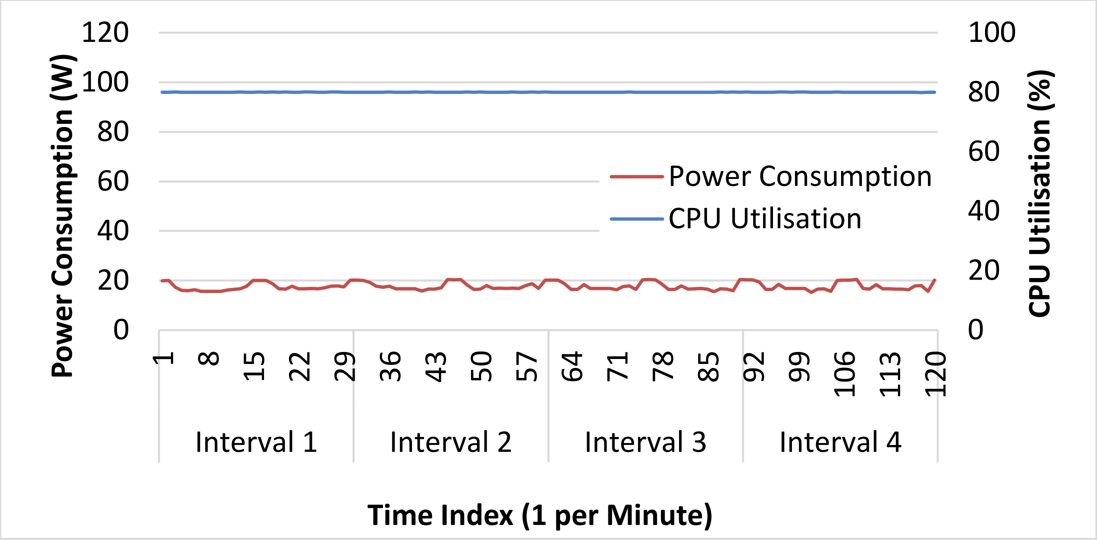

Figure 6-20: Historical Data for a Small VM Based on Static Workload

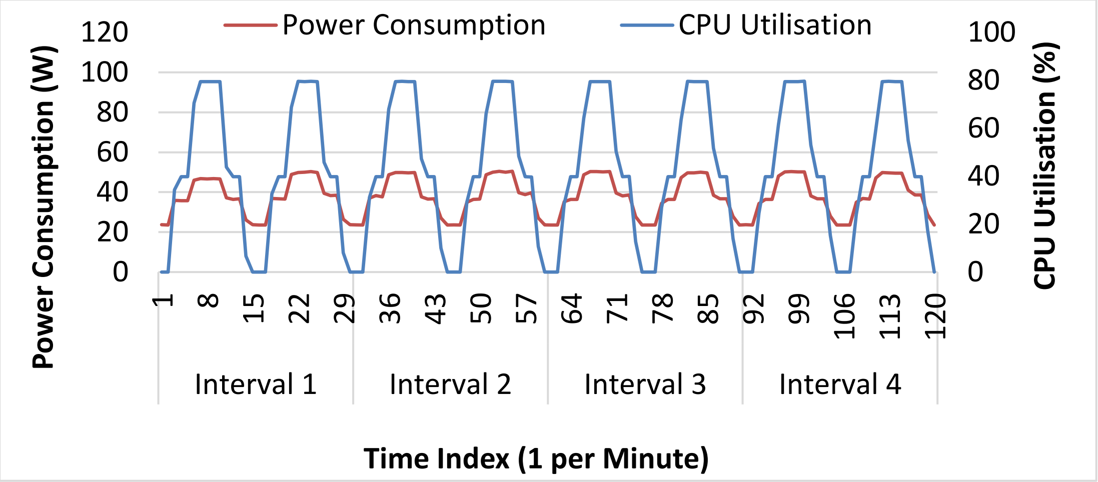

Figure 6-21: Historical Data for a Large VM Based on Periodic Workload

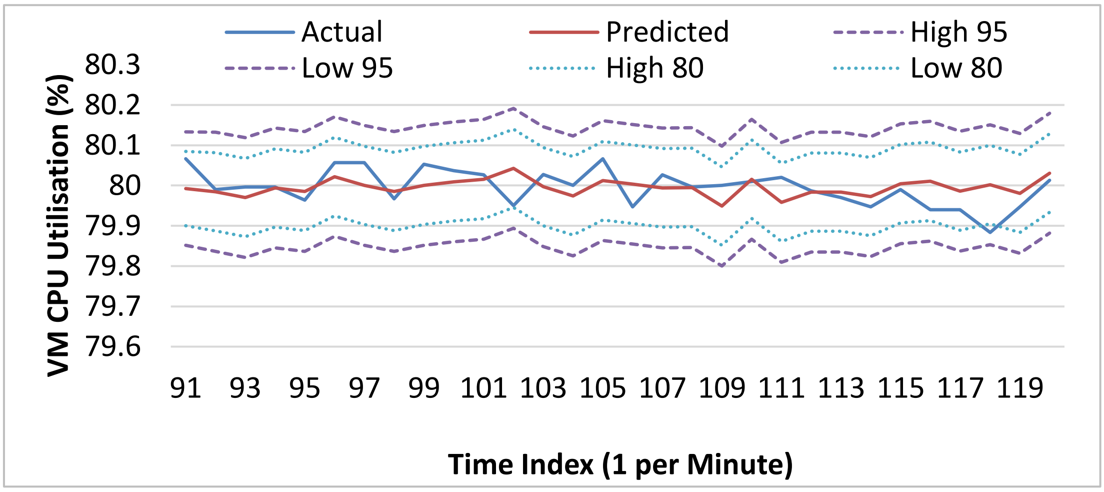

Figure 6-22: Workload Prediction for a Small VM Based on Static Workload Pattern

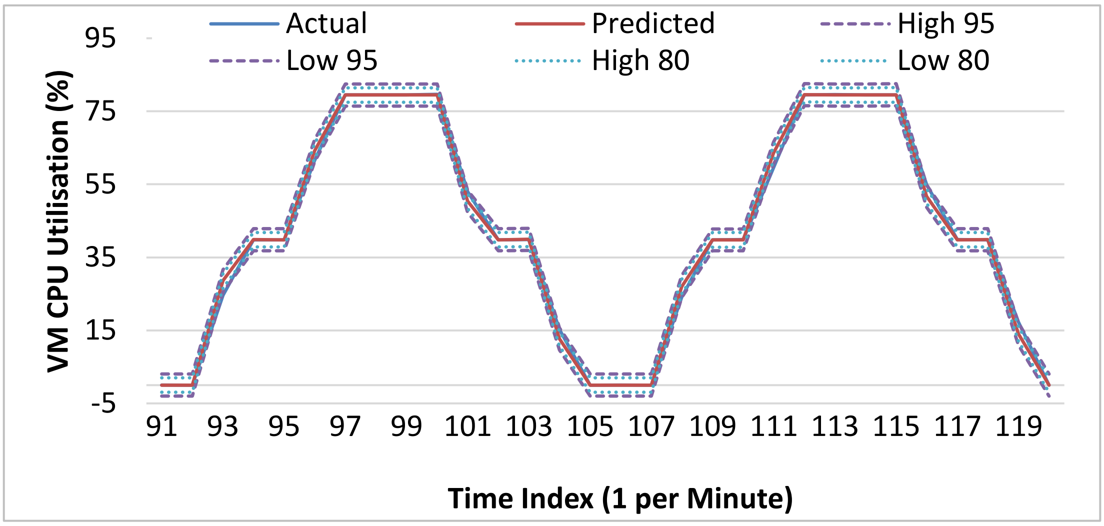

Figure 6-23: Workload Prediction for a Large VM Based on Periodic Workload Pattern

The predicted power consumption results versus the actual values for the small and large VMs are presented on Figures 6-24 and 6-25, respectively. The predicted power consumption of the small VM has a few variations as compared with the actual values. Again, the same reason with the variation of the actual power consumption for the small VM on Host_A is experienced in this small VM.

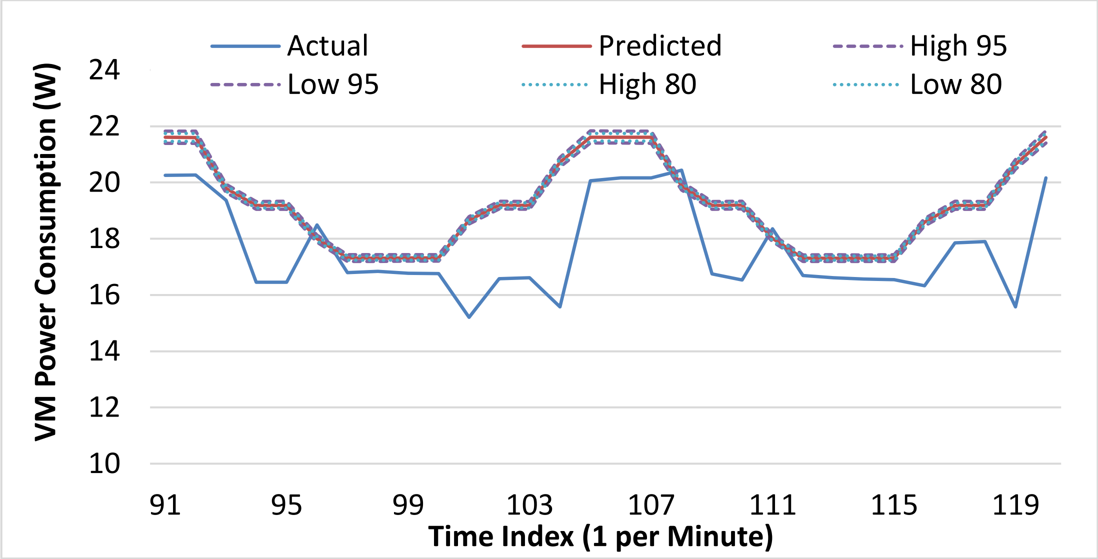

Figure 6-24: Power Prediction for a Small VM Based on Static Workload Pattern

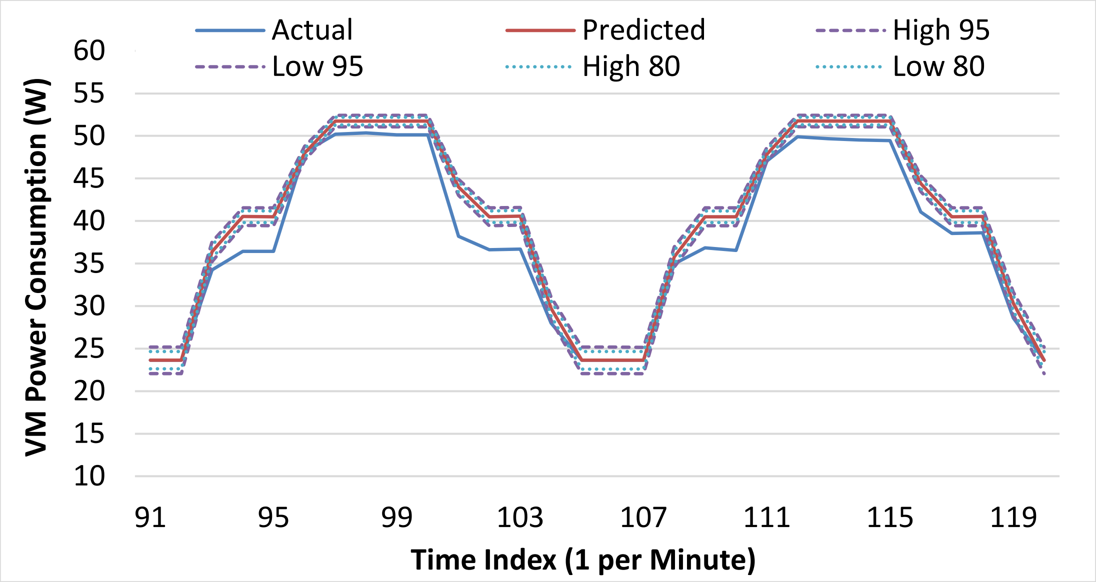

Figure 6-25: Power Prediction for a Large VM Based on Periodic Workload Pattern

The predicted power consumption for the large VM is closely matched with the actual power, as shown on Figure 6-25.

In terms of the prediction accuracy, Table 6-7 shows the evaluation of the predicted results for the small VM, and Table 6-8 shows the evaluation of the predicted results for the large VM.

Table 6-7: Prediction Accuracy for a Small VM Based on Static Workload Pattern

Accuracy Metric	Predicted VM Workload	Predicted VM Power Consumption
ME	0.0005	-1.55
RMSE	0.04	2.08
MAE	0.03	1.65
MPE	0.0006	-9.28
MAPE	0.04	9.76

Table 6-8: Prediction Accuracy for a Large VM Based on Periodic Workload Pattern

Accuracy Metric	Predicted VM Workload	Predicted VM Power Consumption
ME	-0.05	-1.93
RMSE	1.65	2.47
MAE	0.87	1.94
MPE	2.75	-4.96
MAPE	7.09	4.98

The values of the accuracy metrics shown on Tables 6-7 and 6-8 indicate that both VMs achieve a good prediction accuracy in terms of the workload, with 0.0006 of MPE for the small VM and 2.75 for the large VM.

The predicted power consumption results for both of these two types of VMs are a little overestimated when being run on Host_B, as shown on Figures 6-24 and 6-25. When these two types of the VMs being run on Host_A, their power consumption are estimated with almost the same level of the actual power, as shown on Figures 6-18 and 6-19. The main reason of this is because of the underlying Host_B has an overestimation of the actual power consumption while Host_A has the same level of estimation with the actual power. Thus, Tables 6-7 and 6-8 indicate that the predicted power consumption results for small and large VMs when being run on Host_B are less accurate than when being run on Host_A, as shown on Tables 6-5 and 6-6.

6.4.5 Overall Results Discussion

The aim of the proposed energy-aware prediction framework is to address the third research question (**Q.3** – see Section 1.3) by forecasting the power consumption of the VMs prior to service deployment based on historical static and periodic workload data. Based on the results of the conducted experiments, the proposed energy-aware prediction framework is capable of forecasting the workload and the power consumption with a good prediction accuracy for a VM based on historical static workload patterns, as presented in Section 6.4.2, and based on historical periodic workload pattern, as presented in Section 6.4.3.

The third experiment shown in Section 6.4.4 revealed that the framework is also capable of forecasting workload and power consumption for mixed types of VMs, each with a different workload pattern, when being run on two PMs having different characteristics. Furthermore, this experiment has revealed that the predicted power consumption for the same type of VMs when being run on Host_B is less than the predicted power consumption when being run on Host_A since Host_B has less power characteristics in terms of the idle and active as compared to Host_B. Hence, enabling energy-awareness prior to the service deployment can help Cloud service providers to efficiently deploy the VMs on the suitable host that can achieve and maintain the energy goals of the VMs.

6.5 Summary

The proposed energy-aware prediction framework for forecasting the power consumption of VMs prior to service deployment has been presented and discussed comprehensively in this chapter. Also, this has been followed by a demonstration of a number of experiments along with their results to evaluate the capability of the proposed framework for forecasting the power consumption of VMs in future run-time based on historical static and periodic workload patterns running on different types of VMs.

The overall evaluation of the work presented in this thesis will be discussed in the next chapter.

Chapter 7 Thesis Evaluation

7.1 Overview

This chapter provides an overall evaluation of the work proposed in this thesis. It first presents an overview of the research motivation in Section 7.2. After that, an overview of the conducted experiments along with statistical assessment to evaluate their results are presented in Section 7.3. The discussion of the related work is then followed and compared with the work introduced in this thesis, as presented in Section 7.4. Finally, Section 7.5 concludes this chapter by discussing the limitations of the research based on the results obtained from the conducted experiments and the comparison with the related work.

7.2 Motivation

IT services offered by Cloud Computing technology have been widely used by individuals and businesses, especially SMEs. Cloud Computing has increasingly become popular owing to the incentive offered to the consumers to save the capital costs of buying IT resources to just renting the IT resources from the Cloud service providers with affordable costs based on their usage. In this way, the consumers can also save effort and costs to maintain the IT resources as it becomes the Cloud service providers' responsibility.

Cloud Computing infrastructures consist of a large amount of computing resources along with cooling resources that consume a large amount of energy in order to operate. This excessive use of energy has caused ecological issues in terms of the dissipated heat from these resources increasing gas emissions to the atmosphere. This has also caused economic issues especially with the increased pricing of electricity. Hence, the excessive use of energy has been considered a major overhead to maintain by the Cloud service providers [3]. Therefore, managing the energy consumption has become very significant in Cloud Computing environments.

To optimise the energy efficiency in a Cloud environment, energy information is first required. Software analysts of Cloud application would need this information in order to specify energy goals within the requirements to be followed by the software developers when creating and optimising the applications. Further, the developers would need to use programming models that incorporate this information and therefore provide energy-awareness to help them make energy efficient decisions during the construction of the applications. Moreover, energy information is also needed to be incorporated with other tools to help Cloud service providers enhance their decisions when deploying and managing the Cloud services and improving the energy efficiency of their infrastructures. Consequently, profiling and forecasting the energy consumption has become significantly a requirement in order to optimise the energy efficiency in Cloud environments.

Cloud services can run on one or many VMs, which can be hosted on different PMs. The energy consumption of the PMs can be identified easily by using any of the on-the-shelf hardware Watt meters. However, the energy consumption of VMs cannot be identified easily as they do not have a physical hardware interface into which Watt meters can be attached, and it would therefore require modelling the energy consumption of the underlying PMs. Thus, identifying the energy consumption of the VMs has also become critical for Cloud Computing.

7.3 Results Analysis

In this thesis, an energy-aware Cloud system architecture has been proposed in order to enable energy-awareness at the VM level in a Cloud environment. Within this architecture, an energy-aware profiling model has been proposed to profile the energy consumption of the VMs during service operation. Additionally, an energy-aware prediction framework has been proposed to forecast the energy consumption of VMs prior to the service deployment.

Next, Section 7.3.1 will present an overall evaluation of the energy-aware profiling model, and Section 7.3.2 will present an overall evaluation of the energy-aware prediction framework.

7.3.1 Energy-aware Profiling Model

A number of experiments have been demonstrated in Chapter 5 to evaluate the energy-aware profiling model, which is the key element facilitating the EPU component of the proposed architecture as discussed in Section 5.2. The aim of the model is to fairly attribute the PM's energy consumption to homogeneous and heterogeneous VMs based on their CPU utilisation and size in terms of the number of VCPUs each VM has. The overall outcome of the results obtained from these experiments on a Cloud testbed reveals that the proposed model is capable of fairly attributing the PM's energy consumption, including their idle and active energy, to heterogeneous and homogeneous VMs. Further, the results of these experiments reveal that a large part of the VMs energy consumption is based on their idle energy consumption, being attributed form their underlying PM's idle energy. Hence, as it has been also argued in [138], the attribution of the idle energy consumption of the PMs to the VMs is very critical for the Cloud service providers in order to alleviate the costs of the PMs' idle energy.

As the energy consumption of VMs is inferred from the underlying PMs, the total energy consumption of all VMs running on a PM should be equal to the energy consumption of that PM. The work presented in [123] has attributed the PM's active energy only to the VMs. In order to evaluate their profiling tool, the authors added the PM's idle energy to the sum of all VMs' energy consumption and compared it with the total energy consumption of the PM. Likewise, to further evaluate the energy-aware profiling model proposed in this thesis, the total energy consumption of the VMs, including their idle and active energy, is compared with the energy consumption of their underlying PM, as to be presented next for each experiment in Chapter 5.

7.3.1.1 Experiment 1

This experiment, as presented in Section 5.4.2, has shown the results of attributing the energy consumption to two small VMs, representing homogeneous VMs running together on a PM. Figure 7-1 shows the total energy consumption of these two VMs in comparison with the energy consumption of their hosting PM.

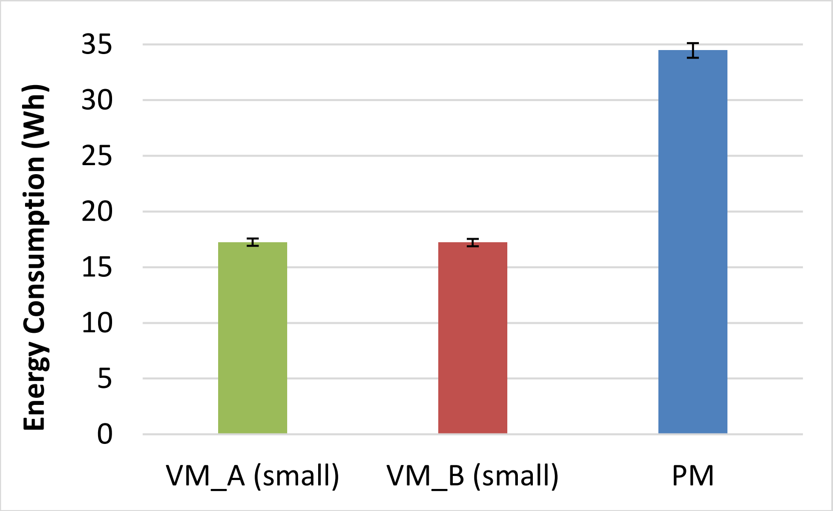

Figure 7-1: All VMs Energy vs PM Energy (for 30 minutes)

As shown in the above Figure 7-1, the energy consumption of VM_A is 17.25 Wh, and 17.21 Wh for VM_B. Despite having the same size and workload, there is a very small variation between the attribution of energy consumption to these two VMs. The main reason of this variation is because of the small variation of their CPU utilisation considered when attributing their active energy. It is important to note that these two small VMs have exactly the same attribution of idle energy consumption as it is based on their size in terms of number of VCPUs only and not considering their CPU utilisation.

By aggregating the energy of these two VMs, their total energy consumption is the same as their underlying PM's energy consumption, which is 34.46 Wh. Thus, the model is capable of attributing the exact PM's energy consumption, including its idle and active energy, to the VMs.

7.3.1.2 Experiment 2

This experiment has shown the results of the energy consumption attribution to a small VM and a large VM, representing heterogeneous VMs running altogether on the same PM, as presented in Section 5.4.3. As depicted on Figure 7-2, the energy consumption of VM_A is 10.60 Wh, and 30.93 for VM_B. VM_B has about three times of energy attribution as compared to VM_A, which corresponds fairly to their size as VM_B is three times larger than VM_A.

By summing up the energy of these two VMs, their total energy consumption is equal to the energy consumption of their hosting PM, which is 41.53 Wh. Hence, the model does not neglect any of the PM's energy consumption when being attributed to heterogeneous VMs.

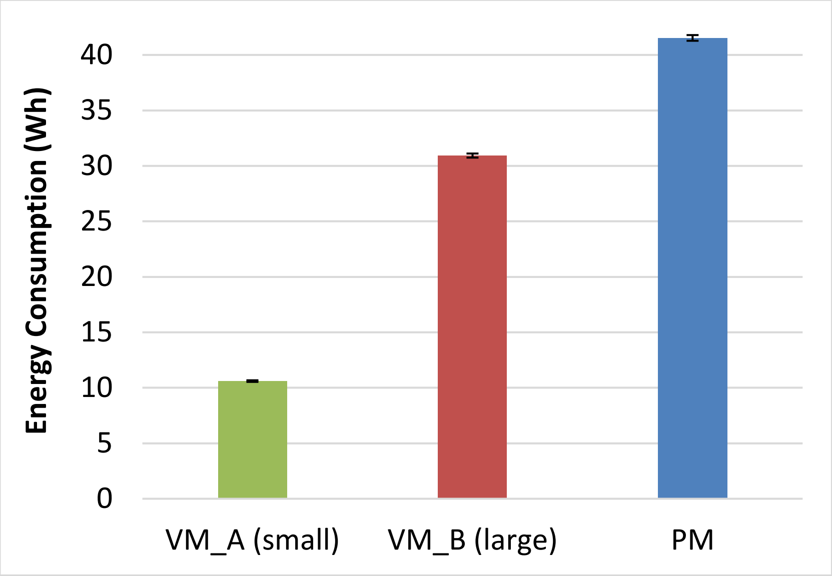

Figure 7-2: All VMs Energy vs PM Energy (for 30 minutes)

7.3.1.3 Experiment 3

As presented in Section 5.4.4, the third experiment has shown the results of attributing the energy consumption to three heterogeneous VMs, small, medium and large, running on a PM, Host_A. Also, this experiment has shown the results of the energy consumption attribution to the same types of these three VMs when running on another PM, Host_B.

Figure 7-3 shows that the energy consumption measured in Wh unit is 7.23 for VM_A, 14.40 for VM_B, and 22.27 for VM_C, which corresponds fairly based on their size. By aggregating the energy consumption of these three VMs, their total energy consumption is 43.90 Wh, which is equal to their underlying PM, Host_A.

When the same type of these three VMs are running on the other PM, Host_B, the energy consumption is 4.02 Wh for VM_A, 8.00 Wh for VM_B, and 11.99 Wh for VM_C. Their total energy consumption is 24.01 Wh, which equally matches the energy consumption of Host_B, as depicted on Figure 7-4.

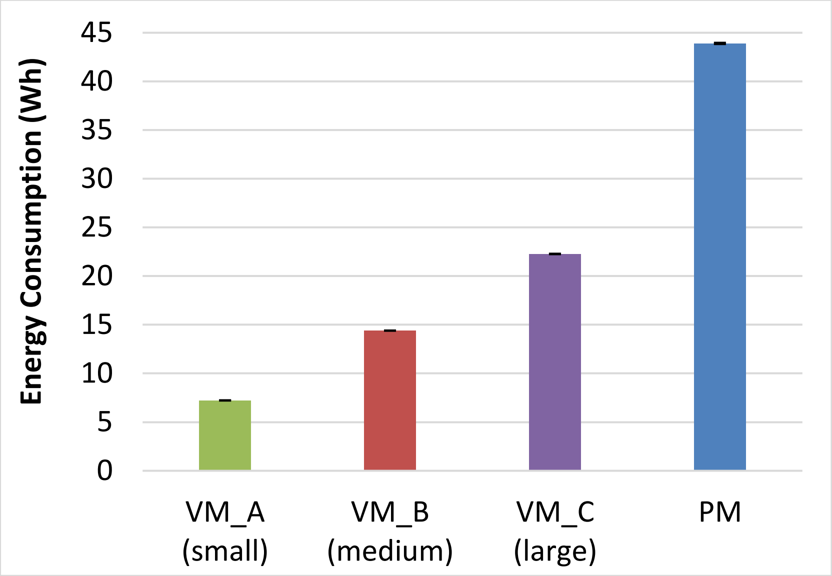

Figure 7-3: All VMs Energy vs PM Energy - Host_A (for 30 minutes)

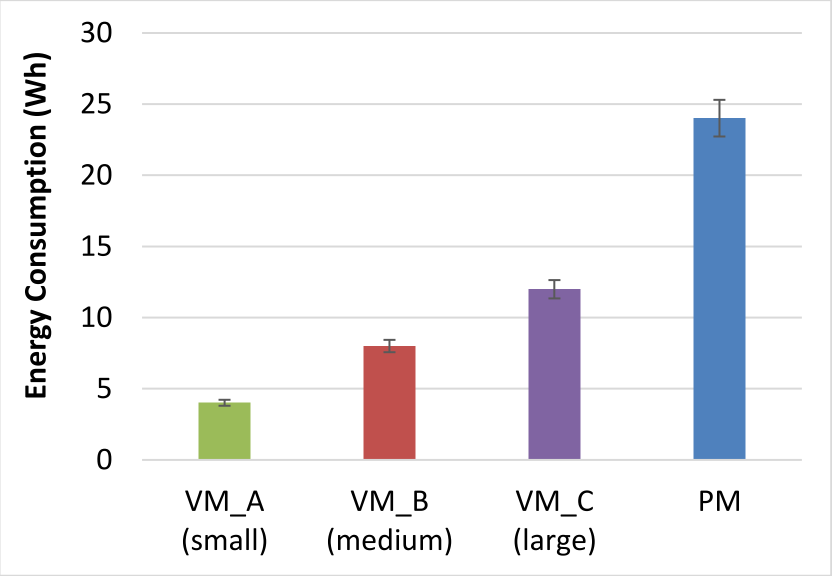

Figure 7-4: All VMs Energy vs PM Energy - Host_B (for 30 minutes)

7.3.2 Energy-aware Prediction Framework

The energy-aware prediction framework proposed in this thesis aims to forecast the power consumption of VMs prior to service deployment based on historical static and periodic workload patterns. This framework is a key part of the EPREU component within the proposed architecture that enables energy prediction of the VMs at the deployment level, as discussed in Section 6.2. A number of direct experiments have been conducted in a Cloud testbed to evaluate the framework, as demonstrated in Chapter 6. The results obtained reveal that the framework is capable of forecasting the workload and power consumption with a good prediction accuracy for a VM prior to the deployment based on historical static and periodic workload patterns. Additionally, the results obtained indicate that the framework is also able to achieve a good prediction accuracy when forecasting the workload and power consumption for different types of VMs with different workload patterns prior to their deployment altogether on two PMs having different characteristics.

Statistical significance tests can be applied using SPSS statistical tool [167] to further evaluate the results obtained. As there are two data groups, predicted and actual, the Two-Sample *T* test (also known as the Independent Samples *T* test) can be used to test whether the mean of the two samples are statistically significantly different or not [168]. Let the null and alternative hypothesis of the *T* test be expressed as follows:

- Null hypothesis (H_0): there is no significant difference statistically between the mean of the two samples, which is supported when the p-value of the *T* test is larger than 0.05.
- Alternative hypothesis (H_1): there is a significant difference statistically between the mean of the two samples, which is supported when the p-value of the *T* test is equal or less than 0.05.

In order to use the *T* test, the data should be normally distributed; otherwise the non-parametric Mann-Whitney U test (MWU), the well-known alternative to the *T* test for non-normal data, can be used to test whether the two samples follow the same distribution or not [169]. Let the null and alternative hypothesis of the MWU test be expressed as follows:

- Null hypothesis (H_0): the two samples statistically significantly follow the same distribution, which is supported when the p-value of the MWU test is larger than 0.05.
- Alternative hypothesis (H_1): the two samples statistically significantly do not follow the same distribution, which is supported when the p-value of the MWU test is equal or less than 0.05.

For data distribution normality testing, the well-known Shapiro-Wilk test can be used [170]. Let the null and alternative hypothesis of the Shapiro-Wilk test be expressed as follows:

- Null hypothesis (H_0): the data is statistically significantly normally distributed, which is supported when the p-value of the Shapiro-Wilk test is larger than 0.05.
- Alternative hypothesis (H_1): the data is statistically significantly not normally distributed, which is supported when the p-value of the Shapiro-Wilk test is equal or less than 0.05.

Next, the obtained predicted results of the VMs' workload and power consumption for the three experiments discussed in Chapter 6 are evaluated in terms of their statistical significance with their corresponding actual data using SPSS with 95% of Confidence Interval. For each experiment, the actual and predicted data distribution is first tested for normality by the Shapiro-Wilk test. Moreover, if both actual and predicted data is statistically significantly normally distributed when their p-values are larger than 0.05, the Independent Sample *T* test is used; otherwise, the alternative non-parametric MWU test is used.

7.3.2.1 Evaluation Results

Table 7-1 presents the results of the statistical significance tests for all three experiment. Beginning with the first experiment, it has shown the predicted workload and power consumption for one large VM prior to deployment on a single PM based on historical static workload pattern, as discussed in Section 6.4.2. As shown in Table 7-1, the null hypothesis of the normality test in this experiment is rejected for both actual and predicted data of the VM workload and power consumption as their p-values of Shapiro-Wilk test are less than 0.05. The alternative hypothesis is supported that the data is statistically significantly not normally distributed. The non-parametric MWU test is then used, and it reveals a p-value larger than 0.05 for both VM workload and power consumption, meaning that null hypothesis is supported and therefore the predicted VM's workload and power consumption statistically significantly have the same distribution as their corresponding actual data.

In terms of the second experiment, it has shown the prediction results of the workload and power consumption for a large VM prior to deployment on a PM based on historical periodic workload pattern, as discussed in Section 6.4.3. The evaluation results of this experiment presented in Table 7-1 show that the actual and predicted data of the VM workload and power consumption are statistically significantly not normally distributed. Therefore, the MWU test is used, and it reveals that the predicted data of the VM workload and power consumption statistically significantly follow the same distribution as their corresponding actual data.

Table 7-1: Evaluation Results of the Energy-aware Prediction Framework

Data	Shapiro-Wilk Test Actual–Predicted Data	*T* Test	MWU Test
Exp. 1			
VM workload	0.004 – 0.001	-	0.104
VM power consumption	0.000 – 0.001	-	0.183
Exp. 2			
VM workload	0.001 - 0.001	-	0.912
VM power consumption	0.001 - 0.001	-	0.128
Exp 3. (Host_A)			
Small VM workload	0.002 - 0.644	-	0.391
Small VM power consumption	0.001 - 0.000	-	0.169
Large VM workload	0.003 - 0.003	-	0.819
Large VM power consumption	0.002 - 0.003	-	0.615
Exp 3. (Host_B)			
Small VM workload	0.395 - 0.995	0.952	-
Small VM power consumption	0.000 - 0.002	-	0.000
Large VM workload	0.003 - 0.002	-	0.690
Large VM power consumption	0.002 - 0.001	-	0.287

As discussed in Section 6.4.4, the third experiment has shown the results of workload and power consumption prediction for two types of VMs, a small and a large, based on static and periodic workload pattern, respectively, prior to deployment on two different PMs, Host_A and Host_B. As shown in Table 7-1, the evaluation results of two VMs (on Host_A) show that the actual and predicted data of the workload and power consumption is statistically significantly not normally distributed. The MWU test is therefore used, and it reveals that predicted data of the two VMs' workload and power consumption statistically significantly have the same distribution as their corresponding actual data.

In terms of the evaluation results of the small VM (on Host_B), the actual and predicted workload data is statistically significantly normally distributed as their p-values of the Shapiro-Wilk test are larger than 0.05. Therefore, the Independent Samples *T* test is used and reveals a p-value larger than 0.05 implying that there is no significant difference statistically between the mean of the predicted workload and the mean of the actual workload. For the power consumption of the small VM, the MWU test is used as the actual and predicted data is not statistically significantly normally distributed. The p-value of the MWU test is less than 0.05 indicating that null hypothesis is rejected and the alternative hypothesis is supported that the distribution of the predicted power consumption statistically significantly has the same shape but it is shifted from the distribution of the actual power consumption; in other words, the results indicate that the data of either the predicted or actual power consumption tends to be larger than the other one. This outcome can be expected as the predicted power is larger than actual power almost across all the interval time, as can be seen in Figure 6-24. Also, the MPE of the predicted VM's power consumption is -9.28, as shown in Table 6-7.

Finally, the evaluation results of the large VM (on Host_B) show that both actual and predicted data of the VM workload and power consumption are statistically significantly not normally distributed. The MWU test is then used and shows that the predicted data of the workload and power consumption for the large VM statistically significantly have the same distribution as their corresponding actual data.

7.3.3 Overall Discussion

All in all, the evaluation of the energy-aware profiling model, as presented in Section 7.3.1, has revealed that the total energy consumption of all VMs in each of the conducted experiments, discussed in Chapter 5, matches their underlying hosting PM. Hence, the proposed model is capable of profiling the exact amount of the PM's energy consumption to the VMs; it neither attributes more energy or less energy to the VMs than the energy consumed by their underlying PM.

In terms of the evaluation of the energy-aware prediction framework, as presented in Section 7.3.2, the predicted workload and power consumption of almost all VMs in each of the experiments, discussed in Chapter 6, equally have the same distribution as their corresponding actual data. The only exception within the obtained evaluation results is for the predicted power consumption of the small VM on Host_B that reveals not having equal distribution as the actual power consumption. As discussed earlier, this outcome corresponds to the observed results in Figure 6-24 that shows the predicted power consumption of this small VM is larger than its actual power for almost the whole interval time.

7.4 Comparison of Research Approaches with Related Work

Enabling energy-awareness at the VM level in Cloud environments has become significant and attracted the attention of many researchers. As discussed in Section 3.3.3, different approaches and models have been introduced to identify the energy consumption of VMs based on the energy consumption of the underlying PMs on which the VMs are running. Table 7-1 presents a comparison of these related energy models along with the model introduced in this thesis for profiling the energy consumption to the VMs.

In terms of the PM’s idle power consumption, as shown on Table 7-2, most of the related work does not consider it or attributes it evenly to the VMs, which would not be fair when heterogeneous VMs are running alongside on the same PM. The only exception is the model presented in [138] which considers attributing the PM’s idle power consumption to homogeneous and heterogeneous VMs; yet when part of the PM’s CPU and memory resources are assigned to the VMs, it only attributes part of the PM’s idle power to VMs, which is considered unfair as that given PM is switched on to run and maintain the status of the VMs; otherwise, that given PM could be switched off to save its idle power consumption. In terms of the PM’s active power consumption, four of the related work models, [123], [134], [138], [140], attribute it to homogeneous VMs only, as shown in Table 7-2. The other models, presented in [130], [141], [142], consider attributing the PM’s active power to homogeneous and heterogeneous VMs, but using different approaches. The model introduced in [130] is the only model that has a similar approach to the one introduced in this thesis when attributing the PM’s active power consumption to the VMs; however, their model still lacks fair attribution of the PM’s idle power consumption to heterogeneous VMs.

Table 7-2: Comparison of VMs Energy-aware Profiling Models

Criteria By	Attributing PM's idle power usage? (Mechanism/Resources)	Attributing PM's active power usage? (Mechanism/Resources)	Type of VMs (Heterogeneous/ Homogeneous)
(Kansal et al, 2010) [123]	Not considered.	Yes. All PM's active power is attributed to VMs based on linear models of PM's power and resources usage, like CPU, memory, and disk, by each VM.	Considered homogeneous VMs only.
(Quesnel et al, 2013) [138]	Yes. Part of the PM's idle power is attributed to VMs based on the assigned PM resources, memory and CPU, and the utilisation of these resources by each VM. If any of the PM's CPU or memory resources is fully assigned to VMs, then all PM's idle power is attributed to VMs.	Yes. All PM's active power is attributed to VMs based on their CPU utilisation.	Considered heterogeneous and homogeneous VMs for idle power, and only homogeneous VMs for the active power.
(Zakarya and Gillam, 2016) [130]	Yes. All PM's idle power is evenly attributed to the running VMs.	Yes. All PM's active power is attributed to VMs based on the allocated physical CPU resources to each VM and the CPU utilisation by each VM.	Considered heterogeneous and homogeneous VMs for active power, but only homogeneous VMs for the idle power.
(Kavanagh et al, 2015) [140]	Yes. All PM's idle power is evenly attributed to the running VMs.	Yes. All PM's active power is attributed to VMs based on their CPU utilisation.	Considered homogeneous VMs only.
(Jiang et al, 2013) [134]	Yes. All PM's idle power is evenly attributed to the running VMs.	Yes. All PM's active power is attributed to VMs based on a two dimensional-LUT that returns a specific power value based on given CPU utilisation and LLC miss rate by each VM.	Considered homogeneous VMs only.
(Chengjian et al, 2013) [141]	Not considered	Yes. All PM's active power is attributed VMs based on performance event counters of CPU and memory components.	Considered homogeneous and heterogeneous VMs for the active power only.
(Yang et al, 2014) [142]	Not considered.	Yes. All PM's active power is attributed to VMs by using SVR model to estimate the power consumption of VMs based on their relationship with the selected performance counters of CPU, memory, disk, cache, process, and network components.	Considered homogeneous and heterogeneous VMs for the active power only.
This Research	Yes. All PM's idle power is fairly attributed to the VMs based on their size in terms of the number of VCPUs.	Yes. All PM's active power is attributed to the VMs fairly based on their CPU utilisation and size.	Considered homogeneous and heterogeneous VMs for the active and idle power.

The energy-aware profiling model presented in this thesis is different when compared to existing models found in the literature. It considers attributing the PM's idle power consumption to heterogeneous and homogeneous VMs based on their size in terms of the number of VCPUs each VM has, which reflects the actual PM's CPU resource and power usage as discussed in Section 5.2.1.1. Also, the PM's active power consumption is attributed to homogeneous and heterogeneous VMs based on their CPU utilisation and size. Thus, the model introduced in this research is the only one that considers homogeneous and heterogeneous VMs when attributing both the idle and active power consumption.

In terms of forecasting the future energy consumption of VMs prior to deployment, it would first require forecasting their workload, which can be then translated into energy based on their physical resource usage. Most of the related work, as discussed in Section 3.3.4, presented different approaches to predict the workload in order to meet the demand and efficiently provision the resources in Cloud environments, yet not considering the energy consumption and energy efficiency of the resources. However, only the work presented in [155] considers predicting the workload and translating it into energy consumption in Cloud environment. The following Table 7-3 presents a comparison of these related work along with the work presented in thesis for forecasting.

Table 7-3: Comparison of Forecasting Approaches

Criteria / By	Predicting workload? (Type of workload)	Predicting Energy? (Level of prediction)
(Patel et al, 2015) [148]	Yes. The considered workload is PM CPU utilisation.	Not considered
(Zhang et al, 2015) [149]	Yes. The considered workload is PM CPU utilisation, network throughput, and data storage size.	Not considered
(Khan et al, 2012) [150]	Yes. The considered workload is VM CPU utilisation.	Not considered
(Fang et al, 2012) [151]	Yes. The considered workload is PM CPU utilisation.	Not considered
(Huang et al, 2013) [153]	Yes. The considered workload is PM CPU utilisation and memory usage.	Not considered
(Calheiros et al, 2015) [154]	Yes. The considered workload is Number of users requests.	Not considered
(Farahnakian et al, 2013) [155]	Yes. The considered workload is PM CPU utilisation.	Yes. Energy prediction at PM level.
This Research	Yes. The considered workload is the VM CPU utilisation and PM CPU utilisation.	Yes. Energy prediction at the PM and VM levels.

As shown in Table 7-3, the work presented in [155] is the only work that has a similar approach to the one introduced in this thesis in terms of forecasting the workload and then translating it into energy consumption. Nonetheless, their approach is only focused at the PM level, whereas the prediction approach introduced in this thesis focuses at both the VM and PM levels. The approach of the framework presented in this thesis first forecasts the workload of the VMs and then correlates the predicted VM workload with the PM to estimate the PM's workload and power consumption, from which the power consumption for the VMs is predicted.

7.5 Limitations

The direct experiments conducted on the Cloud testbed along with their evaluation demonstrate very promising results for enabling energy-awareness at the VM level during the operation and prior to the deployment of the services in Cloud environments. Though, there are a few limitations, as follows:

- The proposed energy-aware profiling model only considers the CPU resource, VM CPU utilisation and number of VCPUS, when profiling the energy to the VMs. Other resources usage such as memory, storage and network throughput are not taken into account. However, many of the related work concluded that the CPU is the only component that affects the power consumption and any other components do not have any impact on the power or indirectly impact the power through triggering the CPU component.
- The proposed energy-aware prediction framework does not consider the thermal energy when predicting the PM's power consumption. Considering the thermal energy can be useful as it can have an effect on the PM's power consumption, as discussed in the results of experiments 1 and 2 demonstrated in Chapter 6. Despite the predicted VM's power consumption in these two experiments achieved high accuracy without consideration of the thermal energy, their prediction could be even more accurate if the thermal energy was considered.
- The step of predicting the VM workload in the proposed energy-aware prediction framework is based on historical static and periodic workload patterns only; other patterns such as once-in-a-lifetime, unpredictable, and continuously changing can be considered. Yet, this is still very

encouraging as being the first work to present VM workload prediction driven through the patterns of Cloud application workload. Additional patterns can be further explored in the future.

7.6 Summary

This chapter has presented an overall evaluation of the work introduced in this thesis. It has firstly revisited the research motivation, then provided an overall evaluation of the conducted experiments along with statistical significance of their results. After that, this chapter has presented a comparison between the related work and the work introduced in this thesis and found that the proposed energy model is the only model that considers both homogeneous and heterogeneous VMs when attributing their idle and active energy. Also, it has found that the prediction framework is the only work that predicts the VM workload and correlates it with the physical resources to predict the PM workload and power consumption, and therefore predict the VM power consumption. This chapter has finally discussed the limitations of the research.

The next chapter will conclude by providing an overall summary of the work presented in this thesis, the main contributions and future work directions.

Chapter 8 Conclusion

This chapter concludes this thesis and provides a summary of the conducted research, as presented in Section 8.1. The key contributions of the research are followed and discussed in Section 8.2. Finally, some future work directions that can be explored based on the work presented in this research are suggested and discussed in Section 8.3.

8.1 Research Summary

Energy consumption has become one of the most important overheads to maintain by the Cloud service providers [3], as it is being extensively consumed to operate the large computing along with the cooling resources in Cloud environments. Consequently, efficiently maintaining and optimising the energy consumption in Cloud Computing environments has increasingly become an important research topic for both academia and industry. In order to optimise the energy efficiency in Cloud environment, energy-awareness should be provided in different layers of Cloud Computing. The software analysts can benefit from energy information to identify energy goals within the requirements of the Cloud application. Incorporating energy information in programming models can help the software developers obtain energy-awareness and enhance their programming decisions while constructing the applications. Further, the Cloud service providers can benefit from obtaining energy information to enhance their decisions to efficiently deploy and manage the Cloud services and improve the energy efficiency of their Cloud resources. Thus, profiling and predicting the energy consumption in Cloud environments has become very critical in order to achieve energy efficiency enhancement of the Cloud resources.

Therefore, the work presented in this thesis aims at enabling energy-awareness in Cloud environment. The energy consumption of the PMs can be easily identified by using any of the hardware Watt meters, but identifying the energy consumption of VMs is challenging and requires the use of the appropriate profiling model that infers their energy from their underlying PMs. Hence, an energy-aware Cloud system architecture is introduced along with a profiling model and a prediction framework to enable energy awareness at the VM level during the operation and deployment of Cloud services.

- **Chapter 2:** introduces the background of the research in terms of Cloud Computing aspects including its definition, services types, deployments types and virtualisation technology. A detailed description of the Cloud system architecture is presented highlighting all main layers along with their roles and interactions. Additionally, properties, design patterns and workload patterns of Cloud applications are discussed. Then, the issues in terms of energy consumption and energy efficiency in Cloud Computing are highlighted along with the streams and trends towards addressing these issues.
- **Chapter 3:** focuses on reviewing the related work towards energy efficient Cloud Computing. It starts by discussing the existing work on energy-aware computing that emphasised the importance of incorporating energy information in different layers of the Cloud stack, e.g. within the requirement engineering to specify energy goals, the programming models to aid the developers with energy-awareness to write efficient code that would consume less energy when operating, and the tools used for deployment and operation of the Cloud services to efficiently manage the resources with energy awareness in mind. In addition, the related work

on profiling and modelling the energy consumption at both PM and VM levels during service operation in Cloud environments is reviewed. Finally, existing models for predicting the workload and energy consumption in Cloud environments are discussed.

- **Chapter 4:** reviews the motivation and importance of energy-awareness in Cloud environments. The proposed energy-aware Cloud system architecture aiming to enable energy-awareness at the VM level is introduced. Detailed descriptions of the architecture's main components along with their roles and how they interact to achieve their objectives are discussed. Some early experiments along with their results are demonstrated to evaluate the ability of the proposed architecture for enabling energy-awareness in a Cloud environment.
- **Chapter 5:** introduces the energy-aware profiling model used as the key part of the proposed Cloud system architecture presented in Chapter 4 for enabling energy-awareness at the VM level during the service operation time. This model focuses on profiling the energy consumption of homogeneous and heterogeneous VMs fairly based on their CPU utilisation and size. A thorough discussion of the development of this model is provided. A number of direct experiments are conducted on a Cloud testbed to evaluate the capability of the model of fairly attributing the PMs' energy consumption to homogeneous and heterogeneous VMs during the operation of Cloud services.
- **Chapter 6:** presents the energy-aware prediction framework used as the key part of the architecture discussed in Chapter 4 for providing energy-awareness at the VM level prior to the deployment of the Cloud services. This framework focuses on predicting the VM workload and then

correlating it with the physical resources to get the predicted power consumption of the VMs prior to deployment. A number of direct experiments on the Cloud testbed are demonstrated along with their results to evaluate the framework in terms of its capability to predict the workload and power consumption of the VMs prior to the deployment of Cloud services.

- **Chapter 7:** provides an overall evaluation of the work presented in this thesis. A summary of the research motivation is first presented. This is followed by an overview of the conducted experiments in Chapters 5 and 6 along with further evaluation and statistical analysis of the obtained results. Furthermore, a comparison of the work introduced in this thesis with the related work is provided along with a discussion in terms of their novelty. Finally, the limitations of the research are discussed.

8.2 Research Contributions

In order to address the research questions of this thesis (see Section 1.3), a number of contributions have been presented in this thesis and they are mainly summarised as follows:

- *An energy-aware Cloud system architecture*. This architecture has been proposed in order to address the first research question (**Q.1**) by enabling energy-awareness in Cloud environments. Two key components, EPU and EPREU, are introduced within this architecture in order to identify the energy usage at the VM level during the operation as well as prior to the deployment of Cloud services. Early results presented in Chapter 4 show that the architecture is capable of identifying the energy consumption of the VMs inferred from the energy consumption of their underlying PMs.

- *An energy-aware profiling model*. This model is the key element of the EPU component within the proposed architecture aiming to enable energy-awareness at the VM level during service operation. As different sizes of VMs can be hosted on the same PM, this model has been developed to address the second research question (**Q.2**) by fairly attributing the PMs' energy consumption to the VMs with consideration of their homogeneity and heterogeneity. The results presented in Chapter 5 show that this model is capable of profiling the PMs' energy consumption to homogeneous and heterogeneous VMs fairly based on their CPU utilisation and size.
- *An energy-aware prediction framework*. This framework is the key element of the EPREU component within the architecture focusing on enabling energy-awareness at the VM level prior to the service deployment. A number of models have been introduced within this framework with the overall objective to address the third research question (**Q.3**) by predicting the power usage of the VMs prior to deployment. Firstly, the VM workload is predicted using ARIMA model based on historical static and periodic workload patterns. Then, the predicted VM workload is correlated with the physical resources using regression models introduced within this framework in order to get the predicted PM's power consumption, based on which the predicted power consumption for the VMs is identified. The results presented in Chapter 6 show that high prediction accuracy of the VMs' power consumption along with their workload has been achieved by the introduced framework.

8.3 Future Work Directions

To further extend the work presented in this thesis, there are some directions that can be followed, as suggested next:

- The energy-aware profiling model presented in this research focuses on enabling energy-awareness for the VM instances in a Cloud environment, exploiting hypervisor-based virtualisation. With the evolving technologies of container-based virtualisation in Clouds [57], [63], a promising extension of the model in that context is to consider attributing the PMs' energy consumption to container instances instead of VM instances. This extension would be useful to enable energy-awareness in Cloud environments not only based on hypervisor-based virtualisation but also on container-based virtualisation.
- The energy-aware prediction framework presented in this thesis does not consider the impact of the thermal energy when predicting the power consumption of PMs. An extension to this is to also consider the thermal energy when correlating the PM's CPU utilisation to the power consumption. This would be a beneficial enhancement which may increase the accuracy of the predicted power consumption of the PMs, and hence the VMs.
- The VM workload prediction within the framework considers only historical static and periodic workload patterns. Another suggested extension is to consider additional Cloud applications workload patterns, e.g. unpredictable, once-in-a-lifetime, and continuously changing. This extension would be valuable to broaden the scope of using the framework

to predict the workload and power consumption of the VMs based on different types of workload patterns.

References

[1] Gartner, “Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions,” *Gartner, Inc.*, 2007. [Online]. Available: http://www.gartner.com/newsroom/id/503867. [Accessed: 01-Oct-2013].

[2] P. Scheihing, “Creating Energy-Efficient Data Centers,” in *Data Center Facilities and Engineering Conference*, 2007.

[3] T. Mukherjee, K. Dasgupta, S. Gujar, G. Jung, and H. Lee, “An economic model for green cloud,” in *Proceedings of the 10th International Workshop on Middleware for Grids, Clouds and e-Science - MGC ’12*, 2012, pp. 1–6.

[4] X. Zhang, J. Lu, and X. Qin, “BFEPM: Best Fit Energy Prediction Modeling Based on CPU Utilization,” *2013 IEEE Eighth Int. Conf. Networking, Archit. Storage*, pp. 41–49, Jul. 2013.

[5] J. Conejero, O. Rana, P. Burnap, J. Morgan, B. Caminero, and C. Carrión, “Analyzing Hadoop power consumption and impact on application QoS,” *Futur. Gener. Comput. Syst.*, vol. 55, pp. 213–223, 2016.

[6] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis, “Energy-Efficient Cloud Computing,” *Comput. J.*, vol. 53, no. 7, pp. 1045–1051, Aug. 2009.

[7] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Y. Zomaya, “A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems,” *CoRR*, vol. abs/1007.0, 2010.

[8] M. Bagein, J. Barbosa, V. Blanco, I. Brandic, S. Cremer, H. D. Karatza, L. Lefevre, T. Mastelic, and A. Oleksiak, “Energy Efficiency for Ultrascale Systems: Challenges and Trends from Nesus Project,” *Supercomput. Front. Innov.*, vol. 2, no. 2, pp. 105–131, Apr. 2015.

[9] J.-J. Jheng, F.-H. Tseng, H.-C. Chao, and L.-D. Chou, “A novel VM workload prediction using Grey Forecasting model in cloud data center,” in *Information Networking (ICOIN), 2014 International Conference on*, 2014, pp. 40–45.

[10] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: state-of-the-art and research challenges,” *J. Internet Serv. Appl.*, vol. 1, no. 1, pp. 7–18, Apr. 2010.

[11] C. Ponsard, R. De Landtsheer, G. Ospina, and J. Deprez, “Towards Design-time Simulation Support for Energy-aware Cloud Application Development,” in *Proceedings of the 6th International Conference on Cloud Computing and Services Science: TEEC*, 2016, vol. 2, no. Closer, pp. 398–404.

[12] C. Xian, Y.-H. Lu, and Z. Li, “A Programming Environment with Runtime Energy Characterization for Energy-Aware Applications,” *Low Power*

Electronics and Design (ISLPED), 2007 ACM/IEEE International Symposium on. pp. 141–146, 2007.

[13] T. Hönig and C. Eibel, “Proactive Energy-Aware System Software Design with SEEP,” in *In Porceedings of 2nd Workshop on Energy-Aware Software-Engineering and Development (EASED@BUIS)*, 2013.

[14] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing,” *Futur. Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.

[15] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage publications, 2013.

[16] Y. Gao, H. Guan, Z. Qi, B. Wang, and L. Liu, “Quality of Service Aware Power Management for Virtualized Data Centers,” *J. Syst. Archit.*, vol. 59, no. 4–5, pp. 245–259, Apr. 2013.

[17] A. I. Avetisyan, R. Campbell, I. Gupta, M. T. Heath, S. Y. Ko, G. R. Ganger, M. A. Kozuch, D. O’Hallaron, M. Kunze, T. T. Kwan, K. Lai, M. Lyons, D. S. Milojicic, H. Y. Lee, Y. C. Soh, N. K. Ming, J. Y. Luke, and H. Namgoong, “Open Cirrus: A Global Cloud Computing Testbed,” *Computer (Long. Beach. Calif).*, vol. 43, no. 4, pp. 35–43, 2010.

[18] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Softw. Pract. Exp.*, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[19] S. Yeo and H. H. S. Lee, “Using Mathematical Modeling in Provisioning a Heterogeneous Cloud Computing Environment,” *{C}omputer*, vol. 44, no. 8, August, pp. 55–62, 2011.

[20] G. Jung, M. a. Hiltunen, K. R. Joshi, R. D. Schlichting, and C. Pu, “Mistral: Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures,” in *2010 IEEE 30th International Conference on Distributed Computing Systems*, 2010, pp. 62–73.

[21] D. J. Armstrong, “Enhancing Quality of Service in Cloud Computing Through Novel Resource Management,” PhD Thesis, University of Leeds, 2012.

[22] P. Garraghan, D. McKee, X. Ouyang, D. Webster, and J. Xu, “SEED: a scalable approach for cyber-physical system simulation,” *IEEE Trans. Serv. Comput.*, vol. 9, no. 2, pp. 199–212, 2016.

[23] R. E. Kavanagh, “Negotiated Resource Brokering for Quality of Service Provision of Grid Applications,” PhD Thesis, University of Leeds, 2013.

[24] D. Chorafas, *Cloud Computing Strategies*. United States of America: CRS Press, Taylor and Francis Group, 2011.

[25] 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,” *Futur. Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, Jun. 2009.

[26] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared,” *Grid Computing Environments Workshop, 2008. GCE ’08*. pp. 1–10, 2008.

[27] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A Break in the Clouds,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, Dec. 2009.

[28] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” *National Institute of Standards and Technology*, 2011. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. [Accessed: 20-Feb-2013].

[29] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, “NIST Cloud Computing Rreference Architecture,” *NIST special publication 500-292*, 2011. [Online]. Available: http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=909505. [Accessed: 09-Jan-2015].

[30] R. Buyya, R. Ranjan, and R. N. Calheiros, “Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities,” *High Performance Computing & Simulation, 2009. HPCS ’09. International Conference on*. pp. 1–11, 2009.

[31] A. Velte, T. Velte, and R. Elsenpeter, *Cloud Computing: A Practical Approach*. United States of America: McGraw-Hill, 2010.

[32] Z. Rehman, F. K. Hussain, and O. K. Hussain, “Towards Multi-criteria Cloud Service Selection,” *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*. pp. 44–48, 2011.

[33] N. Susanta and C. Tzi-Cker, “A survey on virtualization technologies,” *RPE Report*. pp. 1–42, 2005.

[34] N. L. S. da Fonseca and R. Boutaba, “Virtualization in the Cloud,” *Cloud Services, Networking, and Management*. Wiley-IEEE Press, p. 432, 2015.

[35] T. Arthi and H. S. Hamead, “Energy aware cloud service provisioning approach for green computing environment,” *Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference on*. pp. 139–144, 2013.

[36] K. Ye, D. Huang, X. Jiang, H. Chen, and S. Wu, “Virtual Machine Based Energy-Efficient Data Center Architecture for Cloud Computing: A Performance Perspective,” in *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int’l Conference on & Int’l Conference on*

Cyber, Physical and Social Computing (CPSCom), 2010, pp. 171–178.

[37] J. Hardy, L. Liu, N. Antonopoulos, W. Liu, L. Cui, and J. Li, “Assessment and Evaluation of Internet-Based Virtual Computing Infrastructure,” *Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2012 IEEE 15th International Symposium on*. pp. 39–46, 2012.

[38] “The Simplest Cloud Management Experience,” *OpenNebula*. [Online]. Available: http://opennebula.org/. [Accessed: 10-Oct-2016].

[39] “Open Source Software for Creating Private and Public Clouds,” *OpenStack*. [Online]. Available: http://www.openstack.org/. [Accessed: 10-Oct-2016].

[40] “Apache CloudStack^{TM} - Open Source Cloud Computing^{TM},” *Apache CloudStack*. [Online]. Available: http://cloudstack.apache.org/. [Accessed: 10-Oct-2016].

[41] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, “IaaS cloud architecture: From virtualized datacenters to federated cloud infrastructures,” *Computer (Long. Beach. Calif).*, no. 12, pp. 65–72, 2012.

[42] “Distributed Management Task Force - Open Virtualization Format.” [Online]. Available: http://www.dmtf.org/standards/ovf. [Accessed: 01-Oct-2016].

[43] “OpenStack Networking (‘Nova’),” *OpenStack*. [Online]. Available: https://wiki.openstack.org/wiki/Nova. [Accessed: 18-Oct-2016].

[44] “OpenStack Networking (‘Neutron’),” *OpenStack*. [Online]. Available: https://wiki.openstack.org/wiki/Neutron. [Accessed: 18-Oct-2016].

[45] “OpenStack Object Storage (‘Swift’),” *OpenStack*. [Online]. Available: https://wiki.openstack.org/wiki/Swift. [Accessed: 18-Oct-2016].

[46] “OpenStack Identity (‘Keystone’),” *OpenStack*. [Online]. Available: https://wiki.openstack.org/wiki/Keystone. [Accessed: 18-Oct-2016].

[47] “OpenStack Image service (‘Glance’),” *OpenStack*. [Online]. Available: https://wiki.openstack.org/wiki/Glance. [Accessed: 18-Oct-2016].

[48] “OpenStack Dashboard (‘Horizon’),” *OpenStack*. [Online]. Available: https://wiki.openstack.org/wiki/Horizon. [Accessed: 18-Oct-2016].

[49] N. Sabharwal, *Apache CloudStack Cloud Computing*. Packt Publishing Ltd, 2013.

[50] “Welcome to CloudStack Documentation !,” *Apache CloudStack*. [Online]. Available: http://docs.cloudstack.apache.org/. [Accessed: 18-Oct-2016].

[51] A. Vogel, D. Griebler, C. A. F. Maron, C. Schepke, and L. G. Fernandes, “Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack,” in *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 2016, pp.

672–679.

[52] “HPE Helion Eucalyptus - Open Source Hybrid Cloud Software for AWS Users,” *Eucalyptus*. [Online]. Available: www.eucalyptus.com. [Accessed: 19-Oct-2016].

[53] “Nimbus is cloud computing for science,” *Nimbus*. [Online]. Available: http://www.nimbusproject.org/. [Accessed: 19-Oct-2016].

[54] “Open Your Virtual Datacenter,” *oVirt*. [Online]. Available: https://www.ovirt.org/. [Accessed: 19-Oct-2016].

[55] “vSphere and vSphere with Operations Management,” *VMware*. [Online]. Available: http://www.vmware.com/products/vsphere. [Accessed: 19-Oct-2016].

[56] R. Morabito, J. Kjallman, and M. Komu, “Hypervisors vs. Lightweight Virtualization: A Performance Comparison,” *Cloud Engineering (IC2E), 2015 IEEE International Conference on*. pp. 386–393, 2015.

[57] R. Dua, A. R. Raja, and D. Kakadia, “Virtualization vs Containerization to Support PaaS,” *Cloud Engineering (IC2E), 2014 IEEE International Conference on*. pp. 610–614, 2014.

[58] “KVM - Kernel Virtual Machine.” [Online]. Available: http://www.linux-kvm.org/. [Accessed: 01-Oct-2016].

[59] T. Hirt, “Kvm-the kernel-based virtual machine,” *Red Hat Inc*, 2010.

[60] “Xen Project.” [Online]. Available: https://www.xenproject.org/. [Accessed: 01-Oct-2016].

[61] “VMware.” [Online]. Available: http://www.vmware.com/. [Accessed: 01-Oct-2016].

[62] T. Kämäräinen, Y. Shan, M. Siekkinen, and A. Ylä-Jääski, “Virtual machines vs. containers in cloud gaming systems,” in *Network and Systems Support for Games (NetGames), 2015 International Workshop on*, 2015, pp. 1–6.

[63] “LXC - Linux Containers.” [Online]. Available: https://linuxcontainers.org/. [Accessed: 20-Jun-2016].

[64] “Docker.” [Online]. Available: https://www.docker.com/. [Accessed: 01-Oct-2016].

[65] “Cloudfoundry warden manages isolated, ephemeral, and resource controlled environments.” [Online]. Available: https://github.com/cloudfoundry/warden. [Accessed: 01-Oct-2016].

[66] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*. Springer, 2014.

[67] J. Varia, “Architecting for the Cloud: Best Practices,” *Amazon Web*

Services - White Paper, 2011. [Online]. Available: https://media.amazonwebservices.com/AWS_Cloud_Best_Practices.pdf. [Accessed: 01-Sep-2016].

[68] V. Malcher, “Design Patterns in Cloud Computing,” in *2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, 2015, pp. 32–35.

[69] B. D. Martino, G. Cretella, and A. Esposito, “Semantic and Agnostic Representation of Cloud Patterns for Cloud Interoperability and Portability,” in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, 2013, vol. 2, pp. 182–187.

[70] C. Fehling, F. Leymann, J. Rütschlin, and D. Schumm, “Pattern-Based Development and Management of Cloud Applications,” *Futur. Internet*, vol. 4, no. 1, 2012.

[71] B. Di Martino and A. Esposito, “Towards a common semantic representation of design and cloud patterns,” in *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, 2013, p. 385.

[72] S. A. Abtahizadeh, F. Khomh, and Y.-G. Guéhéneuc, “How green are cloud patterns?,” in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, 2015, pp. 1–8.

[73] “Cloud Patterns,” *CloudPatterns.org*. [Online]. Available: http://cloudpatterns.org/. [Accessed: 01-Oct-2016].

[74] “Cloud Computing Patterns.” [Online]. Available: http://www.cloudcomputingpatterns.org. [Accessed: 01-Oct-2016].

[75] C. Fehling, F. Leymann, and R. Retter, “Your Coffee Shop Uses Cloud Computing,” *IEEE Internet Comput.*, vol. 18, no. 5, pp. 52–59, 2014.

[76] G. Cook, T. Dowdall, D. Pomerantz, and Y. Wang, “Clicking clean: how companies are creating the green internet,” 2014.

[77] S. Garg, C. Yeo, and R. Buyya, “Green cloud framework for improving carbon efficiency of clouds,” *Euro-Par 2011 Parallel Process.*, 2011.

[78] M. Dayarathna, Y. Wen, and R. Fan, “Data Center Energy Consumption Modeling: A Survey,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 732–794, 2016.

[79] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing: Foundations and Applications Programming*. Boston: Morgan Kaufmann, 2013.

[80] M. Pawlish, A. S. Varde, and S. A. Robila, “Cloud Computing for Environment-Friendly Data Centers,” in *Proceedings of the fourth international workshop on Cloud data management*, 2012, pp. 43–48.

[81] B. Whitehead, D. Andrews, A. Shah, and G. Maidment, “Assessing the

environmental impact of data centres part 1: Background, energy use and metrics,” *Build. Environ.*, vol. 82, pp. 151–159, Dec. 2014.

[82] K. Bilal, S. U. R. Malik, S. U. Khan, and A. Y. Zomaya, “Trends and Challenges in Cloud Datacenters,” *IEEE Cloud Comput.*, vol. 1, no. 1, pp. 10–20, 2014.

[83] L. Neves, J. Krajewski, P. Jung, and M. Bockemuehl, “GeSI SMARTer 2020: The Role of ICT in Driving a Sustainable Future,” *Global e-sustiainibility initiative (GeSI), Tech. Rep*. 2012.

[84] A. Strategy, “#SMARTer2030: ICT Solutions for 21st Century Challenges,” *The Global eSustainability Initiative (GeSI), Brussels, Brussels-Capital Region, Belgium, Tech. Rep*. 2015.

[85] L. Giese, H. Hausi, M. Mary, S. Rogerio, “10431 Abstracts Collection -- Software Engineering for Self-Adaptive Systems.” Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2008.

[86] N. A. Qureshi and A. Perini, “Engineering Adaptive Requirements,” in *2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, 2009, vol. 2009, pp. 126–131.

[87] N. A. Qureshi, A. Perini, N. A. Ernst, and J. Mylopoulos, “Towards a continuous requirements engineering framework for self-adaptive systems,” *2010 First International Workshop on Requirements@Run.Time*. pp. 9–16, 2010.

[88] V. E. Silva Souza, A. Lapouchnian, W. N. Robinson, and J. Mylopoulos, “Awareness Requirements for Adaptive Systems,” in *Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2011, pp. 60–69.

[89] A. Knauss, D. Damian, X. Franch, A. Rook, H. A. Múller, and A. Thomo, “Acon: A learning-based approach to deal with uncertainty in contextual requirements at runtime,” *Inf. Softw. Technol.*, vol. 70, pp. 85–99, 2016.

[90] J. Shalf, “The Analysis of Impact of Energy Efficiency Requirements on Programming Environments,” in *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, 2012, pp. 920–941.

[91] C. Oriaku and I. A. Lami, “Holistic View Angles of Cloud Computing Services Provisions,” *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on*. pp. 97–105, 2012.

[92] S. Schubert, D. Kostic, W. Zwaenepoel, and K. G. Shin, “Profiling software for energy consumption,” in *Proceedings - 2012 IEEE Int. Conf. on Green Computing and Communications, GreenCom 2012, Conf. on Internet of Things, iThings 2012 and Conf. on Cyber, Physical and Social Computing, CPSCom 2012*, 2012, pp. 515–522.

[93] A. Kansal and F. Zhao, “Fine-grained energy profiling for power-aware application design,” *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 2, p. 26, Aug. 2008.

[94] “Welcome to Apache^{TM} Hadoop®!,” 2011. [Online]. Available: http://hadoop.apache.org/. [Accessed: 18-Feb-2013].

[95] “Windows Azure: Microsoft’s Cloud Platform | Cloud Hosting | Cloud Services,” 2012. [Online]. Available: http://www.windowsazure.com/en-us/. [Accessed: 18-Feb-2013].

[96] “Daytona - Microsoft Research.” [Online]. Available: http://research.microsoft.com/en-us/projects/daytona/. [Accessed: 18-Feb-2013].

[97] J. Ekanayake, “Twister: Iterative MapReduce,” 2009. [Online]. Available: http://www.iterativemapreduce.org/. [Accessed: 18-Feb-2013].

[98] “Manjrasoft - Products,” 2008. [Online]. Available: http://www.manjrasoft.com/products.html. [Accessed: 18-Feb-2013].

[99] “Google App Engine — Google Developers.” [Online]. Available: https://developers.google.com/appengine/?hl=en. [Accessed: 18-Feb-2013].

[100] A. Corradi, M. Fanelli, and L. Foschini, “Increasing Cloud power efficiency through consolidation techniques,” in *2011 IEEE Symposium on Computers and Communications (ISCC)*, 2011, pp. 129–134.

[101] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. V Vasilakos, “Cloud Computing : Survey on Energy Efficiency,” *ACM Comput. Surv.*, vol. 47, no. 2, p. 33:1-33:36, 2014.

[102] D. Kliazovich, P. Bouvry, and S. U. Khan, “DENS: Data Center Energy-Efficient Network-Aware Scheduling,” in *2010 IEEE/ACM Int’l Conference on Green Computing and Communications & Int’l Conference on Cyber, Physical and Social Computing*, 2010, pp. 69–75.

[103] W. Chawarut and L. Woraphon, “Energy-Aware and Real-Time Service Management in Cloud Computing,” *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*. pp. 1–5, 2013.

[104] Y. C. Lee and A. Y. Zomaya, “Energy Efficient Utilization of Resources in Cloud Computing Systems,” *J. Supercomput.*, vol. 60, no. 2, pp. 268–280, Mar. 2010.

[105] A. Tchernykh, L. Lozano, U. Schwiegelshohn, P. Bouvry, J. E. Pecero, S. Nesmachnow, and A. Y. Drozdov, “Online Bi-Objective Scheduling for IaaS Clouds Ensuring Quality of Service,” *J. Grid Comput.*, vol. 14, no. 1, pp. 5–22, 2016.

[106] The Green Grid, “Harmonizing Global Metrics for Data Center Energy

Efficiency,” 2012.

[107] K. Grosskop, “PUE for end users-Are you interested in more than bread toasting?,” *Porc. 2nd Work. Energy-Aware Software-Engineering Dev.*, 2013.

[108] G. A. Brady, N. Kapur, J. L. Summers, and H. M. Thompson, “A case study and critical assessment in calculating power usage effectiveness for a data centre,” *Energy Convers. Manag.*, vol. 76, pp. 155–161, Dec. 2013.

[109] P. Bozzelli, Q. Gu, and P. Lago, “A systematic literature review on green software metrics,” VU University, Amsterdam, 2013.

[110] C. Wilke, S. Götz, and S. Richly, “JouleUnit: A Generic Framework for Software Energy Profiling and Testing,” in *Proceedings of the 2013 Workshop on Green in/by Software Engineering*, 2013, pp. 9–14.

[111] Microsoft, “Virtual Machine Pricing,” *Microsoft*, 2016. [Online]. Available: http://www.windowsazure.com/en-us/pricing/details/virtual-machines/. [Accessed: 19-Jul-2016].

[112] Amazon, “Amazon EC2 Pricing,” *Amazon Web Services*, 2016. [Online]. Available: https://aws.amazon.com/ec2/pricing/. [Accessed: 19-Jul-2016].

[113] P. Berndt and A. Maier, “Towards Sustainable IaaS Pricing,” in *Economics of Grids, Clouds, Systems, and Services SE - 13*, vol. 8193, J. Altmann, K. Vanmechelen, and O. Rana, Eds. Springer International Publishing, 2013, pp. 173–184.

[114] K. Brill, “Understanding The True Cost Of Operating A Server,” *facilitiesnet, November*, 2008. [Online]. Available: http://www.facilitiesnet.com/datacenters/article/ Understanding-the-True-Cost-of-Operating-a-Server--10063.

[115] A. Narayan and S. Rao, “Power-Aware Cloud Metering,” *IEEE Trans. Serv. Comput.*, vol. 7, no. 3, pp. 440–451, 2014.

[116] A. Kostopoulos, E. Agiatzidou, and A. Dimakis, “Energy-Aware Pricing within Cloud Environments,” in *Porceedings of the 13th International Conference on Economics of Grids, Clouds, Systems and Services (GECON’2016), Athens, Greece, 20-22 Sep*, 2016.

[117] M. Aldossary and K. Djemame, “Energy Consumption-based Pricing Model for Cloud Computing,” in *32nd UK Performance Engineering Workshop, UKPEW’2016, Bradford, UK, 8-9 September*, 2016, pp. 16–27.

[118] K. Djemame, D. Armstrong, R. Kavanagh, A. Juan Ferrer, D. Garcia Perez, D. Antona, J.-C. Deprez, C. Ponsard, D. Ortiz, M. Macías Lloret, J. Guitart Fernández, F.-J. Lordan Gomis, J. Ejarque, R. Sirvent Pardell, R. M. Badia Sala, M. Kammer, O. Kao, E. Agiatzidou, A. Dimakis, C. Courcoubetis, and L. Blasi, “Energy efficiency embedded service lifecycle: Towards an energy efficient cloud computing architecture,” in *Joint Workshop*

Proceedings of the 2nd International Conference on ICT for Sustainability 2014, 2014, pp. 1–6.

[119] K. Djemame, R. Kavanagh, D. Armstrong, F. Lordan, J. Ejarque, M. Macias, R. Sirvent, J. Guitart, and R. M. Badia, “Energy Efficiency Support through Intra-Layer Cloud Stack Adaptation,” in *Porceedings of the 13th International Conference on Economics of Grids, Clouds, Systems and Services (GECON’2016), Athens, Greece, 20-22 Sep*, 2016.

[120] “Watts Up? Plug Load Meters.” [Online]. Available: www.wattsupmeters.com/secure/products.php?pn=0. [Accessed: 01-Oct-2014].

[121] “EGM-PWM-LAN: EnerGenie Energy Meter LAN,” *GEMBIRD*. [Online]. Available: http://gmb.nl/item.aspx?id=6736. [Accessed: 20-Oct-2016].

[122] “Kill A Watt,” *P3 International Corporation*. [Online]. Available: http://www.p3international.com/products/p4400.html. [Accessed: 20-Oct-2016].

[123] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. Bhattacharya, “Virtual Machine Power Metering and Provisioning,” in *ACM Symposium on Cloud Computing (SOCC)*, 2010.

[124] G. G. Castañé, A. Núñez, P. Llopis, and J. Carretero, “E-mc2: A Formal Framework for Energy Modelling in Cloud Computing,” *Simul. Model. Pract. Theory*, vol. 39, pp. 56–75, Dec. 2013.

[125] R. Basmadjian, F. Niedermeier, and H. De Meer, “Modelling and analysing the power consumption of idle servers,” in *Proc. of 2nd IFIP Conf. on Sustainable Internet and ICT for Sustainability*, 2012, pp. 1–9.

[126] X. Fan, W.-D. Weber, and L. A. Barroso, “Power Provisioning for a Warehouse-sized Computer,” in *Proceedings of the 34th Annual International Symposium on Computer Architecture*, 2007, pp. 13–23.

[127] C.-H. Lien, Y.-W. Bai, and M.-B. Lin, “Estimation by Software for the Power Consumption of Streaming-Media Servers,” *IEEE Trans. Instrum. Meas.*, vol. 56, no. 5, pp. 1859–1870, Oct. 2007.

[128] W. Dargie, “A stochastic model for estimating the power consumption of a processor,” *Comput. IEEE Trans.*, vol. 64, no. 5, pp. 1311–1322, 2015.

[129] P. Garraghan, I. S. Moreno, P. Townend, and J. Xu, “An Analysis of Failure-Related Energy Waste in a Large-Scale Cloud Environment,” *IEEE Trans. Emerg. Top. Comput.*, vol. 2, no. 2, pp. 166–180, 2014.

[130] M. Zakarya and L. Gillam, “An Energy Aware Cost Recovery Approach for Virtual Machine Migration,” in *Porceedings of the 13th International Conference on Economics of Grids, Clouds, Systems and Services (GECON’2016), Athens, Greece, 20-22 Sep*, 2016.

[131] X. Zhang, J.-J. Lu, X. Qin, and X.-N. Zhao, “A High-Level Energy

Consumption Model for Heterogeneous Data Centers," *Simul. Model. Pract. Theory*, vol. 39, pp. 41–55, Dec. 2013.

[132] M. Tang and S. Pan, "A Hybrid Genetic Algorithm for the Energy-Efficient Virtual Machine Placement Problem in Data Centers," *Neural Process. Lett.*, vol. 41, no. 2, pp. 211–221, 2015.

[133] R. Kavanagh, D. Armstrong, and K. Djemame, "Accuracy of Energy Model Calibration with IPMI," in *Proceedings of the 9th IEEE International Conference on Cloud Computing (CLOUD'2016), June 2016, San Francisco, USA*, 2016.

[134] Z. Jiang, C. Lu, Y. Cai, Z. Jiang, and C. Ma, "VPower: Metering Power Consumption of VM," in *Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on*, 2013, pp. 483–486.

[135] C. Reiss, J. Wilkes, and J. Hellerstein, "Google Cluster-Usage Traces: Format + Schema V2.1," 2014.

[136] "Standard Performance Evaluation Corporation: SPECpower_ssj® 2008." [Online]. Available: http://www.spec.org/power_ssj2008/. [Accessed: 20-Oct-2016].

[137] C. Gu, H. Huang, and X. Jia, "Power Metering for Virtual Machine in Cloud Computing-Challenges and Opportunities," *IEEE Access*, vol. 2, pp. 1106–1116, 2014.

[138] F. Quesnel, H. K. Mehta, and J. M. Menaud, "Estimating the Power Consumption of an Idle Virtual Machine," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, 2013, pp. 268–275.

[139] I. Alzamil, K. Djemame, D. Armstrong, and R. Kavanagh, "Energy-Aware Profiling for Cloud Computing Environments," *Electron. Notes Theor. Comput. Sci.*, vol. 318, pp. 91–108, Nov. 2015.

[140] R. Kavanagh, D. Armstrong, K. Djemame, D. Sommacampagna, and L. Blasi, "Towards an Energy-Aware Cloud Architecture for Smart Grids," in *Economics of Grids, Clouds, Systems, and Services: 12th International Conference, GECON 2015, Cluj-Napoca, Romania, September 15-17*, 2015, pp. 190–204.

[141] W. Chengjian, L. Xiang, Y. Yang, F. Ni, and Y. Mu, "System Power Model and Virtual Machine Power Metering for Cloud Computing Pricing," *Intelligent System Design and Engineering Applications (ISDEA), 2013 Third International Conference on*. pp. 1379–1382, 2013.

[142] H. Yang, Q. Zhao, Z. Luan, and D. Qian, "iMeter: An Integrated VM Power Model Based on Performance Profiling," *Futur. Gener. Comput. Syst.*, vol. 36, pp. 267–286, Jul. 2014.

[143] D. Armstrong, R. Kavanagh, and K. Djemame, “ASCETiC Project: D2.2.2 Architecture Specification - Version 2,” 2014. [Online]. Available: http://www.ascetic-project.eu/sites/default/ascetic/files/content-files/articles/D2.2.2 ASCETiC Architecture Specification.pdf. [Accessed: 01-Mar-2015].

[144] A. Tchernykh, U. Schwiegelsohn, V. Alexandrov, and E. Talbi, “Towards Understanding Uncertainty in Cloud Computing Resource Provisioning,” *Procedia Comput. Sci.*, vol. 51, pp. 1772–1781, 2015.

[145] J. Luis, G. García, R. Yahyapour, and A. Tchernykh, “Load Balancing for Parallel Computations with the Finite Element Method,” vol. 17, no. 3, pp. 299–316, 2013.

[146] A. Tchernykh, J. E. Pecero, A. Barrondo, and E. Schaeffer, “Adaptive energy efficient scheduling in Peer-to-Peer desktop grids,” *Futur. Gener. Comput. Syst.*, vol. 36, pp. 209–220, Jul. 2014.

[147] J. M. Ramírez-Alcaraz, A. Tchernykh, R. Yahyapour, U. Schwiegelshohn, A. Quezada-Pina, J. L. González-García, and A. Hirales-Carbajal, “Job Allocation Strategies with User Run Time Estimates for Online Scheduling in Hierarchical Grids,” *J. Grid Comput.*, vol. 9, no. 1, pp. 95–116, Feb. 2011.

[148] J. Patel, V. Jindal, I.-L. Yen, F. Bastani, J. Xu, and P. Garraghan, “Workload Estimation for Improving Resource Management Decisions in the Cloud,” in *2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems*, 2015, pp. 25–32.

[149] L. Zhang, Y. Zhang, P. Jamshidi, L. Xu, and C. Pahl, “Service Workload Patterns for Qos-Driven Cloud Resource Management,” *J. Cloud Comput.*, vol. 4, no. 23, pp. 1–21, 2015.

[150] A. Khan, X. Yan, S. Tao, and N. Anerousis, “Workload Characterization and Prediction in the Cloud: A Multiple Time Series Approach,” *Network Operations and Management Symposium (NOMS), 2012 IEEE*. pp. 1287–1294, 2012.

[151] W. Fang, Z. Lu, J. Wu, and Z. Cao, “RPPS: A Novel Resource Prediction and Provisioning Scheme in Cloud Data Center,” in *Services Computing (SCC), 2012 IEEE Ninth International Conference on*, 2012, pp. 609–616.

[152] Y. Han, J. Chan, and C. Leckie, “Analysing Virtual Machine Usage in Cloud Computing,” in *Services (SERVICES), 2013 IEEE Ninth World Congress on*, 2013, pp. 370–377.

[153] Q. Huang, S. Su, S. Xu, J. Li, P. Xu, and K. Shuang, “Migration-Based Elastic Consolidation Scheduling in Cloud Data Center,” in *Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on*, 2013, pp. 93–97.

[154] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, “Workload

Prediction Using ARIMA Model and its Impact on Cloud Applications' QoS," *IEEE Trans. Cloud Comput.*, vol. 3, no. 4, pp. 449–458, 2015.

[155] F. Farahnakian, P. Liljeberg, and J. Plosila, "LiRCUP: Linear Regression Based CPU Usage Prediction Algorithm for Live Migration of Virtual Machines in Data Centers," in *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, 2013, pp. 357–364.

[156] ZABBIX, "The Enterprise-Class Monitoring Solution for Everyone." [Online]. Available: http://www.zabbix.com/. [Accessed: 01-Jun-2014].

[157] "Software testing as a Service," 2013. [Online]. Available: http://parsa.epfl.ch/cloudsuite/cloud9.html. [Accessed: 01-Sep-2014].

[158] L. Ciortea, C. Zamfir, S. Bucur, V. Chipounov, and G. Candea, "Cloud9: A Software Testing Service," *ACM SIGOPS Oper. Syst. Rev.*, vol. 43, no. 4, pp. 5–10, Jan. 2010.

[159] I. S. Moreno, "Characterizing and Exploiting Heterogeneity for Enhancing Energy-Efficiency of Cloud Datacenters," PhD Thesis, University of Leeds, 2014.

[160] A. Waterland, "Stress Project," 2014. [Online]. Available: http://people.seas.harvard.edu/~apw/stress/. [Accessed: 01-Oct-2015].

[161] KVM -, "Kernel-based Virtual Machine." [Online]. Available: http://www.linux-kvm.org/. [Accessed: 01-Nov-2015].

[162] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, 4th ed. Hoboken, N. J.: John Wiley & Sons, 2008.

[163] R Core Team, "R: A Language and Environment for Statistical Computing," *R Foundation for Statistical Computing*, 2015. [Online]. Available: https://www.r-project.org/. [Accessed: 20-May-2015].

[164] G. E. P. Box and D. R. Cox, "An analysis of transformations," *J. R. Stat. Soc. Ser. B*, pp. 211–252, 1964.

[165] R. J. Hyndman and A. B. Koehler, "Another Look at Measures of Forecast Accuracy," *Int. J. Forecast.*, vol. 22, no. 4, pp. 679–688, Oct. 2006.

[166] R. J. Hyndman and G. Athanasopoulos, "Section 2.5: Evaluating Forecast Accuracy," in *Forecasting: Principles and Practice*, OTexts, 2014.

[167] IBM Corp., "Released 2015. IBM SPSS Statistics for Windows, Version 23." IBM Corp., Armonk, NY.

[168] S. L. Jackson, *Research Methods and Statistics: A Critical Thinking Approach*. Cengage Learning, 2015.

[169] A. Gold, "Understanding the Mann-Whitney Test," *J. Prop. Tax Assess. Adm.*, vol. 4, no. 3, p. 55, 2007.

[170] A. Field, *Discovering Statistics Using IBM SPSS Statistics*. Sage, 2013.