Performance and Energy-based Cost Prediction Modelling of Virtual Machines in Cloud Computing Environments

By

Mohammad Mubark M Aldossary

Submitted in accordance with the requirements

for the degree of Doctor of Philosophy

The University of Leeds

School of Computing

November 2019

Declaration

The candidate confirms that the work submitted is his 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.

Journal Articles:

1. ***Aldossary, M*** *and Djemame, K* (2019) **A Hybrid Approach for Performance and Energy-based Cost Prediction of Virtual Machines in Cloud Computing Environments**. Journal of Systems and Software, Oct 2019, (Submitted). This paper is the candidate's own work. It was reviewed by the co-author Karim Djemame. The content of this paper is included in Chapter 6.
2. ***Aldossary, M****, Djemame, K, Alzamil, I, Kostopoulos, A, Dimakis, A and Agiatzidou, E* (2019) **Energy-Aware Cost Prediction and Pricing of Virtual Machines in Cloud Computing Environments**. Future Generation Computer Systems (FGCS'2019), Vol 93, Apr 2019. The candidate designed the energy-aware cost prediction framework and contributed to the paper's structure. The paper was reviewed by the co-author Karim Djemame who also contributed to the System Architecture. The co-author Ibrahim Alzamil collaborated on the paper with the introduction of the energy-aware virtual machine model. The co-authors Alexandros Kostopoulos, Antonis Dimakis and Eleni Agiatzidou contributed the energy-aware pricing scheme. The contribution of the candidate is included throughout the thesis, mainly in Chapter 4.

Refereed Conference Papers:

3. ***Aldossary, M*** *and Djemame, K* (2018) **Energy-based Cost Model of Virtual Machines in a Cloud Environment**. The 5th International Symposium on Innovation in Information and Communication Technology (ISIICT'2018), 31 Oct - 01 Nov 2018, Philadelphia University, Amman, Jordan. This paper is the candidate's own work. It was reviewed by the co-author Karim Djemame. Its content is included in Chapter 3.
4. ***Aldossary, M*** *and Djemame, K* (2018) **Performance and Energy-based Cost Prediction of Virtual Machines Auto-Scaling in Clouds**. The 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA'2018), 29 - 31 Aug 2018, Prague, Czech Republic, pp. 502–509. This paper is the candidate's own work. It was reviewed by the co-author Karim Djemame. Its content is included in Chapter 5.
5. ***Aldossary, M*** *and Djemame, K* (2018) **Performance and Energy-based Cost Prediction of Virtual Machines Live Migration in Clouds**. The 8th International Conference on Cloud Computing and Services Science, (CLOSER'2018), 19 - 21 Mar 2018, Funchal, Madeira – Portugal, pp. 384–391. This paper is the candidate's own work. It was reviewed by the co-author Karim Djemame. Its content is included in Chapter 5.
6. ***Aldossary, M****, Alzamil, I and Djemame, K* (2017) **Towards Virtual Machine Energy-Aware Cost Prediction in Clouds**. The 14th International Conference on Economics of Grids, Clouds, Systems and Services (GECON'2017), 19 - 21 Sep 2017, Montaury Campus, Anglet, France, pp. 119–131. Most of this paper's content is the candidate's own work. The co-author Ibrahim Alzamil 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, mainly in Chapter 4.
7. ***Aldossary, M*** *and Djemame, K* (2016) **Energy Consumption-based Pricing Model for Cloud Computing**. The 32nd UK Performance Engineering Workshop (UKPEW'2016), 08 - 09 Sep 2016, Bradford, United Kingdom, University of Bradford, pp. 16–27. This paper is the candidate's own work. It was reviewed by the co-author Karim Djemame. The content of this paper is included in Chapter 3.

Acknowledgements

First and foremost, I thank God (*Allah*) for gave me the health, strength and patience to complete this thesis.

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

I also would like to acknowledge and thank Dr Richard Kavanagh for his help and technical support on the Cloud testbed. A special thanks to my colleague Dr Ibrahim Alzamil for his assistance during the early stages of my study. I also want to extend my gratitude to all friends, colleagues, and to the members of the Distributed System and Services Research Group for their valued discussions and support.

My deepest gratitude goes to my parents, Mubark and Fatima, for their enduring love and support throughout my life. Additionally, I express my gratefulness to my beloved wife Afnan for her love and being always there supportive and patient with this long journey. Thanks also to my brothers and sisters Haya, Nourah, Misfer, Ali, Reem, Hassan and Raghad for their support and encouragement at all times.

Finally, I would like to thank Prince Sattam Bin Abdulaziz University for granting the scholarship to do my PhD study in the UK.

Abstract

Cloud Computing has transformed the way in which enterprises and individuals are utilising the Information Technology (IT) by offering on-demand services such as applications, platforms and infrastructures for their customers with reasonable prices based on their usage (e.g., *pay-as-you-go* model). However, the wide adoption of Cloud Computing and the growing number of Cloud customers have increased the overall operational costs for Cloud providers, especially with the increasing cost of energy consumed to operate Cloud services. Consequently, Cloud providers consider energy consumption as one of the most important cost factors to be maintained within their infrastructures.

In order to achieve energy efficiency and reduce the operational costs for Cloud services, *reactive* and *proactive* management mechanisms can be used to efficiently manage Cloud resources and reduce energy-related costs while maintaining service performance requirements. However, these mechanisms need to be supported with performance and energy awareness not only at the physical machine (PM) level but also at virtual machine (VM) level in order to make enhanced cost decisions. Moreover, estimating the future cost of Cloud services can help the cloud service providers offer suitable services that meet their customers' requirements.

This thesis introduces a Cloud system architecture along with a novel *Cost Modeller* component that aims to enable the awareness of energy consumption, performance variation and cost in a Cloud environment. To fulfil this aim, an energy-based cost model is firstly developed to attribute the PM's energy consumption to VMs and measures the actual resource usage, power consumption and the total cost for each VM. An energy-based cost prediction framework is then introduced to predict workload, power consumption and estimate the total cost of the VMs during service operation based on historical workload data. Finally, a performance and energy-based cost prediction framework is introduced to combine VMs consolidation and resource provisioning in order to design cost-effective strategies while taking into consideration the trade-off among cost, energy efficiency and performance variation of Cloud services.

The evaluation of the proposed research on a Cloud testbed shows that the proposed energy-based cost model is capable of fairly attributing the PMs energy consumption to heterogeneous VMs, thus enabling cost and energy awareness at the VM level. Compared with actual results obtained in the Cloud testbed, the predicted results show that the proposed energy-based cost prediction framework is capable of predicting workload, power consumption and estimating the total cost for heterogeneous VMs based on historical workload patterns. Additionally, the results have shown that the proposed performance and energy-based cost prediction framework is capable to estimate the total cost of heterogeneous VMs by considering their resource usage and power consumption, while maintaining the expected level of service performance.

The application of the proposed research provides the awareness of energy consumption, performance variation and cost at the virtual level in Cloud environments, which contributes to overcoming the challenge of identifying the most cost-effective strategies for Cloud services. The outcomes of this research can be used and incorporated in *reactive* and *proactive* management mechanisms to make enhanced cost decisions supported by performance and energy awareness in order to efficiently manage Cloud resources. This has the potential to contribute to a reduction in energy consumption, and therefore lowering the total cost for Cloud providers while maintaining the service performance.

List of Abbreviations

AIC	Akaike Information Criterion
ANN	Artificial Neural Network
AR	Auto Regression
ARIMA	Auto-Regressive Integrated Moving Average
ARMA	Auto-Regressive Moving Average
AWS	Amazon Web Services
BIC	Bayesian Information Criterion
CLI	Command Line Interface
CPUs	Central Processing Units
DNN	Deep Neural Network
EC2	Elastic Compute Cloud
ETS	Exponential Smoothing
FPGAs	Field Programmable Gate Arrays
GPUs	Graphic Processing Units
IaaS	Infrastructure as a Service
IM	Infrastructure Manager
IPMI	Intelligent Platform Management Interface
IT	Information Technology
KVM	Kernel-based Virtual Machine
kWh	Kilowatt-Hour
LXC	Linux Containers
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

ME	Mean Error
MI	Monitoring Infrastructure
ML	Machine Learning
MPE	Mean Percentage Error
NFS	Network File System
NIST	National Institute of Standards and Technology
OS	Operating System
PaaS	Platform as a Service
PM	Physical Machine
PVM	Privileged Virtual Machine
QoS	Quality of Service
RAM	Random Access Memory
RAPL	Running Average Power Limit
RMSE	Root Mean Squared Error
SaaS	Software as a Service
SLA	Service Level Agreement
SLO	Service Level Objective
SSD	Solid-State Drive
vCPU	Virtual CPU
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VMM	Virtual Machine Monitor or Manager
W	Watt
Web UI	Web User Interface
Wh	Watt-Hour
XaaS	Everything as a Service

Table of Contents

Declaration ..... **ii**
Acknowledgements ..... **v**
Abstract ..... **vi**
List of Abbreviations ..... **viii**
Table of Contents ..... **x**
List of Figures ..... **xiv**
List of Tables ..... **xviii**
Chapter 1. Introduction **1**
1.1 Motivation 1
1.2 Aim and Objectives 3
1.3 Methodology 6
1.4 Main Contributions 7
1.5 Thesis Overview 8
Chapter 2. Background and Literature Review **11**
2.1 Overview 11
2.2 Cloud Computing 11
2.2.1 Definition 12
2.2.2 System Architecture 13
2.2.3 Services Types 15
2.2.4 Deployment Types 17
2.2.5 Virtualisation 18
2.2.5.1 Virtual Infrastructure Manager 19
2.2.5.2 Hypervisors 20
2.2.5.3 Containers 21
2.3 Cloud Computing Applications 21
2.3.1 Workload Patterns 21
2.3.2 Benchmarking 23
2.4 Pricing Models in Cloud Computing 24
2.5 Energy-related Cost Issues in Cloud Computing 26
2.5.1 Cost Models 27
2.5.2 Cost and Energy Consumption Models 29
2.5.3 Overall Discussion 31
2.6 Prediction Models in Cloud Computing 32
2.6.1 Workload Prediction 33

2.6.2 Energy Prediction 35
2.6.3 Cost Estimation 37
2.6.4 Overall Discussion 39
2.7 Dynamic Resource Management in Cloud Computing 41
2.7.1 VM Consolidation 41
2.7.2 Resource Provisioning 47
2.7.3 Overall Discussion 50
2.8 Thesis Scope 53
2.9 Summary 54
Chapter 3. System Architecture and Energy Cost Modelling 55
3.1 Overview 55
3.2 Definitions and Assumptions 55
3.3 Proposed System Architecture 57
3.3.1 Key Components and Interactions 58
3.3.1.1 SLA Manager 58
3.3.1.2 VM Manager 58
3.3.1.3 Infrastructure Manager 59
3.3.1.4 Monitoring Infrastructure 59
3.3.1.5 Cost Modeller 59
3.4 Energy-based Cost Model 60
3.5 Early Implementation 64
3.5.1 Cloud Testbed 64
3.5.2 Monitoring Infrastructure 65
3.5.3 Specifications of PMs and VMs 66
3.6 Experiments and Evaluation 67
3.6.1 Design of Experiments 67
3.6.2 Evaluation 68
3.7 Summary 73
Chapter 4. Energy-based Cost Prediction Framework 75
4.1 Overview 75
4.2 Energy-based Cost Prediction Framework 75
4.2.1 VMs Workload Prediction 76
4.2.2 PMs Workload Prediction 78
4.2.3 PMs Power Consumption Prediction 79
4.2.4 VMs Power Consumption Prediction 80
4.2.5 VMs Total Cost Estimation 81

4.3 Implementation 81
4.3.1 Characterisation of Physical Machines 82
4.4 Experiments and Evaluation 82
4.4.1 Design of Experiments 82
4.4.2 Evaluation 83
4.5 Summary 89
Chapter 5. Performance and Energy-based Cost Prediction Framework 91
5.1 Overview 91
5.2 Performance and Energy-based Cost Prediction Framework 91
5.2.1 VMs Live Migration Prediction Models 93
5.2.2 VMs Auto-Scaling Prediction Models 100
5.3 Implementation 107
5.3.1 Characterisation of Physical Machines 108
5.4 Experiments and Evaluation 108
5.4.1 Design of Experiments 108
5.4.2 Evaluation 109
5.4.2.1 VMs Workload Prediction 109
5.4.2.2 VMs Power Consumption Prediction 114
5.4.2.3 VMs Total Cost Estimation 117
5.5 Summary 120
Chapter 6. A Hybrid Approach for Performance and Energy-based Cost Prediction 122
6.1 Overview 122
6.2 Integration of VMs Auto-Scaling with Live Migration: A Hybrid Approach 122
6.3 Implementation 131
6.3.1 Characterisation of Physical Machines 131
6.4 Experiments and Evaluation 132
6.4.1 Design of Experiments 132
6.4.2 Evaluation 133
6.4.2.1 VMs Workload Prediction 133
6.4.2.2 VMs Power Consumption Prediction 136
6.4.2.3 VMs Total Cost Estimation 142
6.5 Summary 144
Chapter 7. Conclusion 146
7.1 Research Summary 146

7.2 Research Outcomes.....149
7.2.1 Energy-based Cost Model.....149
7.2.2 Energy-based Cost Prediction Framework.....150
7.2.3 Performance and Energy-based Cost Prediction Framework 151
7.2.4 A Hybrid Approach for Performance and Energy-based Cost Prediction.....152
7.2.5 Comparison of Research Approaches with Related Work ..153
7.3 Research Contributions.....157
7.4 Limitations.....159
7.5 Future Work Directions.....160
References.....162

List of Figures

Figure 2-1: NIST Cloud Computing Reference Architecture Model [34]. 13
Figure 2-2: Layered Cloud Computing Architecture [35]. 14
Figure 2-3: Cloud Computing Architecture [36]. 15
Figure 2-4: Types of Cloud Deployment Models [44]. 17
Figure 2-5: Cloud Application Workload Patterns [66]. 22
Figure 2-6: Thesis Scope. 53
Figure 3-1: System Architecture. 57
Figure 3-2: Cloud Testbed Architecture. 65
Figure 3-3: Monitoring Infrastructure. 66
Figure 3-4: The Workload Results for Small VM (for 30 minutes). 69
Figure 3-5: Power Consumption Small VM on Host A and Host B. 69
Figure 3-6: The Workload Results for Medium VM (for 30 minutes). 70
Figure 3-7: Power Consumption Medium VM on Host A and Host B. 70
Figure 3-8: The Workload Results for Large VM (for 30 minutes). 71
Figure 3-9: Power Consumption Large VM on Host A and Host B. 71
Figure 3-10: PM Mean Power Consumption Attributed to each VM - Host A. 72
Figure 3-11: PM Mean Power Consumption Attributed to each VM - Host B. 72
Figure 3-12: Mean Energy Consumption per VM (for 30 minutes) - Host A. 72
Figure 3-13: Mean Energy Consumption per VM (for 30 minutes) - Host B. 72
Figure 3-14: The VMs Total Cost on Host A and Host B. 73
Figure 3-15: The VMs Cost Saving on Host B. 73
Figure 4-1: Energy-based Cost Prediction Framework. 76
Figure 4-2: Number of vCPUs vs CPU Utilisation for Host A. 78
Figure 4-3: Number of vCPUs vs CPU Utilisation for Host B. 78
Figure 4-4: CPU Utilisation vs Power Consumption for Host A. 79
Figure 4-5: CPU Utilisation vs Power Consumption for Host B. 79
Figure 4-6: The Workload Prediction for Small VM (for 30 minutes). 84
Figure 4-7: The Workload Prediction for Medium VM (for 30 minutes). 85
Figure 4-8: The Workload Prediction for Large VM (for 30 minutes). 86
Figure 4-9: Predicted Small VM Power Consumption. 87
Figure 4-10: Predicted Medium VM Power Consumption. 87
Figure 4-11: Predicted Large VM Power Consumption. 88
Figure 4-12: The Estimated VMs Total Cost on Host A and Host B. 88
Figure 4-13: The Estimated VMs Cost Saving on Host B. 89
Figure 5-1: Performance and Energy-based Cost Prediction Framework. 92
Figure 5-2: Performance and Energy-based Cost Prediction Framework (Live Migration). 94

Figure 5-3: Number of vCPUs (VMx) vs PM CPU Utilisation (Source PMi), Host A. 97
Figure 5-4: Number of vCPUs (VMx) vs PM CPU Utilisation (Destination PMj - most energy efficient PM), Host B. 97
Figure 5-5: Number of vCPUs (VMx) vs PM CPU Utilisation (Destination PMj - less energy efficient PM), Host D. 97
Figure 5-6: The PM CPU Utilisation vs Power Consumption (Source PMi), Host A. 98
Figure 5-7: The PM CPU Utilisation vs Power Consumption (Destination PMj - most energy efficient PM), Host B. 98
Figure 5-8: The PM CPU Utilisation vs Power Consumption (Destination PMj - less energy efficient PM), Host D. 98
Figure 5-9: Performance and Energy-based Cost Prediction Framework (Auto-Scaling). 101
Figure 5-10: The Process of VM Auto-Scaling (Vertical Scaling vs Horizontal Scaling). 104
Figure 5-11: The Workload Prediction Results for Small VM. 110
Figure 5-12: The Workload Prediction Results for Medium VM. 111
Figure 5-13: The Workload Prediction Results for Large VM. 112
Figure 5-14: The Predicted Workload vs The Actual Workload for both PMs (Source PMi and Destination PMj). 113
Figure 5-15: Small VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj). 115
Figure 5-16: Medium VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj). 115
Figure 5-17: Large VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj). 115
Figure 5-18: Small VM Predicted vs Actual Power Consumption using a Predefined VM Size - Scaling on a Number of PMs. 116
Figure 5-19: Small VM Predicted vs Actual Power Consumption using Self-Configuration VM Size - Scaling on a Number of PMs. 116
Figure 5-20: Medium VM Predicted vs Actual Power Consumption using a Predefined VM Size - Scaling on a Number of PMs. 116
Figure 5-21: Medium VM Predicted vs Actual Power Consumption using Self-Configuration VM Size - Scaling on a Number of PMs. 116
Figure 5-22: Large VM Predicted vs Actual Power Consumption using a Predefined VM Size - Scaling on a Number of PMs. 117
Figure 5-23: Large VM Predicted vs Actual Power Consumption using Self-Configuration VM Size - Scaling on a Number of PMs. 117

Figure 5-24: Estimated Total Cost Before vs After Migration with Migration Cost Recovery on (most energy efficient PM), Host B. 118
Figure 5-25: Estimated Total Cost Before vs After Migration with Migration Cost Recovery on (less energy efficient PM), Host D. 118
Figure 5-26: The Potential Migration Cost Recovery on (most energy efficient PM), Host B. 118
Figure 5-27: The Potential Migration Cost Recovery on (less energy efficient PM), Host D. 118
Figure 5-28: Estimated Small VM Auto-Scaling Total Cost (Predefined VM Size Scaling vs Self-Configuration VM Size Scaling). 119
Figure 5-29: Estimated Medium VM Auto-Scaling Total Cost (Predefined VM Size Scaling vs Self-Configuration VM Size Scaling). 119
Figure 5-30: Estimated Large VM Auto-Scaling Total Cost (Predefined VM Size Scaling vs Self-Configuration VM Size Scaling). 119
Figure 5-31: Cost Saving by Self-Configuration Scaling (Small VM). 120
Figure 5-32: Cost Saving by Self-Configuration Scaling (Medium VM). 120
Figure 5-33: Cost Saving by Self-Configuration Scaling (Large VM). 120
Figure 6-1: A Hybrid Approach for Performance and Energy-based Cost Prediction. 124
Figure 6-2: The Workload Prediction Results for Small VM. 134
Figure 6-3: The Workload Prediction Results for Medium VM. 135
Figure 6-4: The Workload Prediction Results for Large VM. 136
Figure 6-5: Small VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj). 137
Figure 6-6: Medium VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj). 137
Figure 6-7: Large VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj). 137
Figure 6-8: Small VM Predicted vs Actual Power Consumption using Vertical Scaling on a Number of PMs. 139
Figure 6-9: Medium VM Predicted vs Actual Power Consumption using Vertical Scaling on a Number of PMs. 139
Figure 6-10: Large VM Predicted vs Actual Power Consumption using Vertical Scaling on a Number of PMs. 139
Figure 6-11: Small VM Predicted vs Actual Power Consumption using Migration and Vertically Scaling on a Number of PMs. 140
Figure 6-12: Medium VM Predicted vs Actual Power Consumption using Migration and Vertically Scaling on a Number of PMs. 140

Figure 6-13: Large VM Predicted vs Actual Power Consumption using Migration and Vertically Scaling on a Number of PMs. 140
Figure 6-14: Small VM Predicted vs Actual Power Consumption using Horizontal Scaling on a Number of PMs. 141
Figure 6-15: Medium VM Predicted vs Actual Power Consumption using Horizontal Scaling on a Number of PMs. 141
Figure 6-16: Large VM Predicted vs Actual Power Consumption using Horizontal Scaling on a Number of PMs. 141
Figure 6-17: Estimated Total Cost Before vs After Migration on Different PMs. 142
Figure 6-18: Estimated Cost Saving for Migrating Small VM to Different PMs. 142
Figure 6-19: Estimated Vertical Scaling VMs Total Cost. 143
Figure 6-20: Estimated Migration and Vertically Scaling VMs Total Cost. 143
Figure 6-21: Estimated Horizontal Scaling VMs Total Cost. 143

List of Tables

Table 2-1: Comparison of Open Source Cloud Platforms. 19
Table 2-2: Summary of Existing Cost and Energy Models. 32
Table 2-3: Summary of Prediction Models. 40
Table 2-4: Summary of Existing Models for VMs' Consolidation and Resource Provisioning. 52
Table 2-5: Summary of Prediction Models for VMs' Consolidation and Resource Provisioning. 52
Table 3-1: Configurations of the PMs. 66
Table 3-2: Configurations of the VMs. 67
Table 4-1: Prediction Accuracy for Small VM. 84
Table 4-2: Prediction Accuracy for Medium VM. 85
Table 4-3: Prediction Accuracy for Large VM. 86
Table 4-4: Prediction Accuracy for The Predicted Power Consumption for all VMs on (Host A and Host B). 88
Table 5-1: Summary of Notations. 95
Table 5-2: Prediction Accuracy for Small VM. 110
Table 5-3: Prediction Accuracy for Medium VM. 111
Table 5-4: Prediction Accuracy for Large VM. 112
Table 5-5: Prediction Accuracy for The Predicted Power Consumption for all VMs on (Host A, Host B and Host D). 115
Table 5-6: Prediction Accuracy The Predicted Power Consumption for all VMs on (Host A, Host B, Host C and Host D). 117
Table 6-1: Prediction Accuracy for Small VM. 134
Table 6-2: Prediction Accuracy for Medium VM. 135
Table 6-3: Prediction Accuracy for Large VM. 136
Table 6-4: Prediction Accuracy for The Predicted Power Consumption for all VMs on Source (Host A) and Destination (Host B, Host C and Host D). 138
Table 6-5: Prediction Accuracy for The Predicted Power Consumption for all VMs performs (Vertical Scaling) on Different Hosts. 139
Table 6-6: Prediction Accuracy for The Predicted Power Consumption for all VMs performs (Migration and Vertically Scaling) on Different Hosts. 140
Table 6-7: Prediction Accuracy for The Predicted Power Consumption for all VMs performs (Horizontal Scaling) on Different Hosts. 141
Table 7-1: Comparison of Cost and Energy Models. 153
Table 7-2: Comparison of Prediction Approaches. 154
Table 7-3: Comparison of Prediction Models for VMs' Consolidation and Resource Provisioning. 156

Chapter 1. Introduction

1.1 Motivation

Cloud Computing is an important and growing business model that has revolutionised the Information Technology (IT) industry by providing different services such as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) for the Cloud customers with reasonable prices based on their usage (e.g., *pay-as-you-go* model). However, the wide adoption of Cloud Computing and the growing number of Cloud customers have increased the overall operational costs for Cloud providers [1]–[5]. Thus, reducing the operational costs of different Cloud services is an active area of research.

The cost mechanisms that are employed by different Cloud service providers significantly influence the adoption of Cloud Computing within the IT industry. In this regards, the cost mechanisms that are offered by Cloud service providers have become sophisticated, as customers are charged per month, hour, minute or second based on the resources they utilise [6]–[8]. Nevertheless, there are still limitations, as customers are charged based on pre-defined tariffs for the resources they utilise. These pre-defined tariffs do not consider the variable cost of energy [9], [10]. With the increasing cost of electricity, Cloud providers consider energy consumption as one of the biggest operational cost factors to be managed within their infrastructures [1]–[3], especially with the large fluctuations in electricity prices [11]. Consequently, modelling a new cost mechanism for Cloud services that can be adjusted to the energy costs has attracted the attention of many researchers [1]–[3].

A number of mechanisms have been adopted by Cloud service providers in order to achieve economies of scale in a Cloud environment. For example, dynamic consolidation presents a solution to improve resource utilisation and achieve energy efficiency in Clouds. Virtual Machines (VMs) consolidation allows VMs to move from one physical machine (PM) to another through live migration, without any interruption in the service. This mechanism plays an important role in load balancing among the PMs and reduces the overall energy consumption

by switching off the idle hosts. However, VMs live migration is a resource-intensive operation which has an impact on the performance of the migrating VM and consequently on the services running on other VMs [12], [13]. Besides, there are additional costs [14] in terms of migration time and energy overhead that need further consideration [15], [16]. Thus, understanding the impact of VM live migration is essential to design a cost-effective VM consolidation strategy. Resource provision defined as VMs auto-scaling is another solution to provide additional capacity to the VMs on-the-fly in order to handle service performance variations. However, this mechanism may take a few minutes to initiate [17], [18], which is unacceptable for VMs that need to rapidly scale during the computation [19], [20]. Also, there are additional costs [14] in terms of scaling time (booting/rebooting), license fees for the new VMs (horizontal scaling) and energy overhead that need attention [21]. Hence, understanding the impact of VMs auto-scaling is important to design a cost-effective resource provision technique.

In addition, most of the existing studies in the literature have focused on minimising the energy consumption and maximising the resource utilisation, instead of improving the performance of applications. To illustrate, Cloud providers such as Amazon EC2 [22] have established their Service Level Agreements (SLAs) based on service availability without such an assurance of the service performance [23]. For instance, during service operation, consider the situation where a number of VMs are running on the same PM, and each VM is allocated its fair share of resources. If the VM's workload increases and no resources are available to handle this increment (e.g., the workload exceeds the acceptable level of Central Processing Unit (CPU) such as 95% threshold), resource competition may occur leading to VMs' performance degradation which may affect the fulfilment of the SLAs and hence the cloud infrastructure provider's revenue. Hence, to prevent such performance loss, *proactive* frameworks have the advantage of taking preventive actions (e.g., auto-scaling, live migration or both) at an early stage to avoid service performance degradation. The effectiveness of such frameworks will depend on potential actuators/decisions to implement at service operation. Furthermore, estimating the future cost of cloud services can help the service providers offer suitable services that meet their customers' requirements.

1.2 Aim and Objectives

This research is aimed towards enabling the awareness of energy consumption, performance variation and cost at the virtual level in Cloud Computing environments, which contributes to overcoming the challenge of identifying the most cost-effective strategies for Cloud services.

The research presented in this thesis requires a number of stages to meet the aims and objectives of this work. The first stage is exploring the issues of the current cost models in Cloud Computing and the identification of a research opportunity, which is the need for enabling the awareness of energy consumption, performance variation and cost at VM level. The next stage is to introduce a *Cost Modeller* as a solution within the system architecture to fulfil this need, followed by the development of an energy-based cost model to attribute the PM's energy consumption to VMs and measure the actual resource usage, power consumption and the total cost for each VM. After that, the energy-based cost prediction framework is introduced to predict workload, power consumption and estimate the total cost of the VMs. The final stage is the introduction of a performance and energy-based cost prediction framework that combines VMs consolidation and resource provisioning in order to design cost-effective strategies while taking into consideration the trade-off between cost, energy efficiency and performance variation in a Cloud environment.

The outcomes of this research can be used and incorporated by *reactive* and *proactive* resource management techniques to make enhanced cost decisions supported by performance and energy awareness in order to efficiently manage the Cloud resources.

Therefore, the following research questions need to be addressed:

- ***Q.1:*** How can a cost model that considers power consumption as a key parameter be established? And what will be the impact of its adoption on Cloud provider's revenue?
- **Q.2:** How can a model that predicts resource usage, power consumption, and estimates the cost for heterogeneous VMs at service operation be

designed? And what will be the impact of enabling cost and energy awareness at the VM and PM levels on the Cloud provider's revenue?

- **Q.3:** How can a prediction cost model that adapts to a performance variation at the PM and VM levels be designed? And what will be the impact of such a model on energy consumption and the total cost of Cloud services?
- **Q.4:** *Based on the predicted results*. How to efficiently get the service/application performance to the expected level with minimal impact on cost? And what will be the impact of VMs consolidation and resource provisioning on the total cost of Cloud services?
- **Q.5:** How can a proactive prediction framework that integrates VMs consolidation with resource provisioning into a hybrid approach be designed? And what will be the impact of such an integration on the cloud provider's revenue?

In order to address these questions, a number of objectives are identified:

- **O.1:** *Exploring the current cost models related issues and challenges in the Cloud paradigm*. Optimising cost mechanisms of Cloud services has been an active research area, especially with the trade-off between cost, energy efficiency and performance variation in Clouds. Therefore, it is essential to understand the current challenges in order to contribute to a solution that can be used to address these challenges.
- **O.2:** *Investigating how the cost models of Cloud services are used in a Cloud environment as well as the identification of their limitations*. Energy consumption is one of the important parameters that influence the cost of Cloud services. The power consumption at the PM level can be easily identified but is not directly measured at the VM level. Thus, understanding how the physical resources are correlated with the virtual resource's usage and their impact on energy consumption is important. This work therefore introduces and implements an energy-based cost model that can fairly attribute PM's energy consumption to VMs and

estimate the actual cost for heterogeneous VMs by considering their resource usage and power consumption.

- **O.3:** *Exploring the use of statistical techniques and prediction methods along with mathematical modelling in order to predict workload, energy consumption and estimate the total cost of the VMs*. This work introduces an energy-based cost prediction framework to predict workload, energy consumption and estimate the total cost for heterogeneous VMs at service operation based on historical time-series workload patterns.
- **O.4:** *Investigating the issues related to VMs consolidation and resource provisioning in a Cloud environment in terms of performance variation and energy consumption*. Thus, understanding the impact of VMs consolidation and resource provisioning is essential to design cost-effective strategies for Cloud services. A set of algorithms that deal with VMs consolidation and resource provisioning are proposed with the aim to minimise the overall costs incurred by the performed decisions. This work introduces a performance and energy-based cost prediction framework that aims to estimate the total cost of heterogeneous VMs by considering their resource usage and power consumption, while maintaining the expected level of service performance.
- **O.5:** *Integrating VMs auto-scaling with dynamic VMs allocation into a hybrid approach in this research context*. A set of algorithms that detect the underloaded and overloaded hosts in order to perform the most cost-effective decision(s) to handle the service performance variation are proposed. This work introduces a hybrid approach for performance and energy-based cost prediction that aims to integrate VMs auto-scaling with live migration. This is aimed at minimising the overall costs incurred by the performed decisions and estimating the total cost of heterogeneous VMs by considering their resource usage and power consumption, while maintaining the expected level of service performance.

1.3 Methodology

In order to achieve the aims and objectives of this research, a quantitative approach with three traditional research methods are used [24]:

- ***Direct Experiments*** [25], [26]: in the context of this research, this method can be defined as conducting direct experiments to validate a hypothesis or a solution on a real Cloud environment, (e.g., a Cloud testbed), which can give most accurate and reliable results. However, this method can be time-consuming to conduct such repeatable experiments and limited to the resources availability. Therefore, it can be hard and costly to conduct large-scale experiments in a real Cloud environment [27].
- ***Mathematical Modelling*** [12], [28]: this method can be defined as a precise formulation of mathematical models that can be idealised or modelled (under a set of assumptions) to match the original system. The models achieved by this method can be validated with a direct implementation in a real environment or in a simulation [29].
- ***Simulation*** [27], [30]: this method can be defined as a simulated use of a real system for conducting experiments to validate a hypothesis or a solution. This method can be easily repeatable and scalable in a controllable environment with low cost. However, simulation involves some randomness that gives less accuracy and reliability as compared to direct experiments. Therefore, simulation alone requires further verification (e.g., combined with mathematical models or direct implementation) in order to represent a real environment [31].

In the context of this research, both mathematical modelling and direct experiments methods are used. Mathematical modelling is used to formulate the energy-based cost model and the prediction models presented in this thesis. Direct experiments are also used and conducted on a Cloud testbed to verify and validate the capability of these models in a real Cloud environment. Furthermore, these research methods will be useful when collecting and analysing the data obtained from a local Cloud testbed, in order to identify relevant metrics and their relationship.

The simulation method has not been considered in this thesis for the following reasons. Firstly, the experimental results achieved from the simulation can be less accurate as compared to the direct experiment. Secondly, it is difficult to understand the real behaviour and correlation of the Cloud resources using simulation. For example, the direct experiments that are conducted in this thesis on a Cloud testbed have helped to identify the required parameters and their correlations for the implementation of the mathematical models, as to be presented in Sections 3.3 and 4.2. However, the simulation method can be considered as future work to further study the scalability-related issues, which is difficult to address with the limited resources in a local Cloud testbed.

1.4 Main Contributions

The main contributions of this thesis are the following:

- *A Cloud system architecture along with an energy-based cost model*. This architecture includes the required components to support energy awareness, performance variation and cost of Cloud services. *The Cost Modeller* is the main architectural component including the other contributions of this thesis. *An energy-based cost model* is established to address the first research question (**Q.1**) by enabling cost and energy-awareness at the VM level. This model can fairly attribute the PM's energy consumption to heterogeneous VMs and measure the actual resource usage, power consumption and the total cost for each VM.
- *An energy-based cost prediction framework*. This framework consists of a number of mathematical models with the aim of addressing the second research question (**Q.2**) by predicting the workload, power consumption and estimating the total cost of the VMs during service operation. This framework makes use of a prediction model for predicting the VMs' workload based on historical workload patterns and correlating the predicted VMs workload with physical resources to predict the power consumption for each VM. The total cost of the VMs' is then estimated based on the predicted VMs workload and power consumption.

- *A performance and energy-based cost prediction framework*. This framework aims to address the third and fourth research questions (**Q.3** and **Q.4**) by estimating the total cost of heterogeneous VMs, considering their resource usage and power consumption, while maintaining the expected level of service performance. This framework includes two approaches that can be used for VMs consolidation and resource provisioning in order to design cost-effective strategies and prevent performance loss at different levels. A set of algorithms have been developed for VMs consolidation and resource provisioning to achieve cost savings while meeting the performance objectives. This framework works by predicting the workload, power consumption and estimating the total cost of the migrated and scaled VMs during service operation based on historical workload data.
- *A hybrid approach for performance and energy-based cost prediction*. This approach aims to address the fifth research question (**Q.5**) by integrating auto-scaling with live migration in order to estimate the total cost of heterogeneous VMs by considering resource usage and power consumption. A set of algorithms have been developed for VMs consolidation and resource provisioning to achieve cost savings while meeting the performance objectives. This approach works by detecting the underloaded and overloaded hosts in order to perform the most cost-effective decision(s) to handle the service performance variation.

1.5 Thesis Overview

The remaining chapters of this thesis are organised as follows:

- **Chapter 2** presents an overview of the fundamental concepts of Cloud Computing, Cloud applications and their workload patterns as well as related benchmarks. A description of Cloud Computing pricing models is also presented. This is followed by positioning the work in the relevant literature, focusing on the energy-related cost issues, prediction models and resource management in Cloud Computing, along with a presentation of the thesis scope.

- **Chapter 3** introduces the system architecture with thorough details of its main components and their interactions. This is followed by a presentation of an energy-based cost model that considers energy consumption as a key parameter. The experiments are performed to evaluate the ability of the proposed system architecture in terms of supporting cost and energy awareness at the VM level in a Cloud environment.
- **Chapter 4** proposes an energy-based cost prediction framework which consists of a number of mathematical models in order to estimate the total cost of VMs by considering the resource usage and power consumption. This is followed by a demonstration of experiments on the Cloud testbed to evaluate the capability of the proposed framework.
- **Chapter 5** introduces a performance and energy-based cost prediction framework that aims to estimate the total cost of VMs by considering their resource usage and power consumption, while maintaining the expected level of service performance. This framework includes two approaches that can be used for VMs consolidation and resource provisioning in order to design cost-effective strategies and prevent performance loss at different levels. A number of algorithms have been developed for VMs consolidation and resource provisioning to achieve cost savings while meeting the performance objectives. This is followed by experiments on a Cloud testbed to evaluate the capability of the proposed framework to predict live migration and auto-scaling total cost for heterogeneous VMs at service operation.
- **Chapter 6** presents a hybrid approach for a performance and energy-based cost prediction. This approach supports decision-making by integrating auto-scaling with live migration, considering their costs, while at the same time being aware of the impact on other quality characteristics such as energy consumption and performance of the application. A number of algorithms have been developed for VMs consolidation and resource provisioning to achieve cost savings while meeting the performance objectives. This is followed by a demonstration of experiments on the Cloud testbed to evaluate the capability of the presented approach to identify the most suitable cost-effective decision(s)

to handle the service performance variation at both physical and virtual levels.

- **Chapter 7** summarises the work, research outcomes and contributions presented in this thesis. This is followed by a discussion of the limitations and future work directions that could further improve this research.

Chapter 2. Background and Literature Review

2.1 Overview

This chapter presents the essential background and reviews the literature on the subject of energy-related cost issues, prediction models and resource management in Cloud Computing. It starts by introducing the fundamental concepts of Cloud Computing with a detailed description of its definition, system architecture, services types, deployment types and virtualisation technologies, as presented in Section 2.2. The aspects of Cloud applications and their workload patterns as well as related benchmarks are discussed in Section 2.3. A description of Cloud Computing pricing models is presented in Section 2.4. This is followed by positioning the work in the relevant literature, focusing on the energy-related cost issues, prediction models and resource management in Cloud Computing. The energy-related cost issues are highlighted, along with a detailed discussion of the closely related work in Section 2.5. It then discusses the prediction models related to the workload, energy consumption and cost of Cloud services, as well as a summarised discussion of the closely related work, as presented in Section 2.6. It also reviews the existing work on dynamic resource management, including VMs consolidation and resource provisioning, along with a summarised discussion of the closely related work, as presented in Section 2.7. Finally, the thesis scope is presented in Section 2.8.

2.2 Cloud Computing

Cloud Computing is a technology that uses the Internet to provide computing resources as services. This innovation allows scalable, on-demand sharing of resources and their costs between Cloud customers. Also, it provides customers with various online computing services at reasonable prices, to manage, process, and store their data efficiently. With the cloud, customers do not need to install any kind of software on their machines; as long as the Internet connection is accessible, they can reach their data worldwide from any computer [32].

In the following subsections, a definition of Cloud Computing, system architecture, service types, deployment types, virtualisation, Cloud applications patterns and pricing models will be discussed.

2.2.1 Definition

Cloud Computing is defined by the National Institute of Standards and Technology (NIST) as:

> "*a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*" p.2, [33].

According to the NIST definition, there are five main characteristics of Cloud Computing [33], [34]:

- ***On-demand self-service****:* the ability of Cloud providers to provision computing resources to their customers as needed without requiring human interaction.
- ***Broad network access****:* through standard mechanisms, the cloud customers can access their resources over the network.
- ***Resource pooling****:* the cloud providers have a pool of computing resources to serve different customers using (e.g., a multi-tenant model).
- ***Rapid elasticity****:* the capacity of Cloud resources can be more flexible and rapidly provisioned.
- ***Measured service****:* the resource utilisation is monitored, automatically measured and optimised.

2.2.2 System Architecture

NIST [33] has presented a high-level system architecture that involves all Cloud actors along with their distinct roles and interactions in Cloud Computing. This Cloud Computing reference architecture model consists of five actors, namely Cloud consumer, Cloud auditor, Cloud provider, Cloud broker and Cloud carrier [35], as depicted in Figure 2-1.

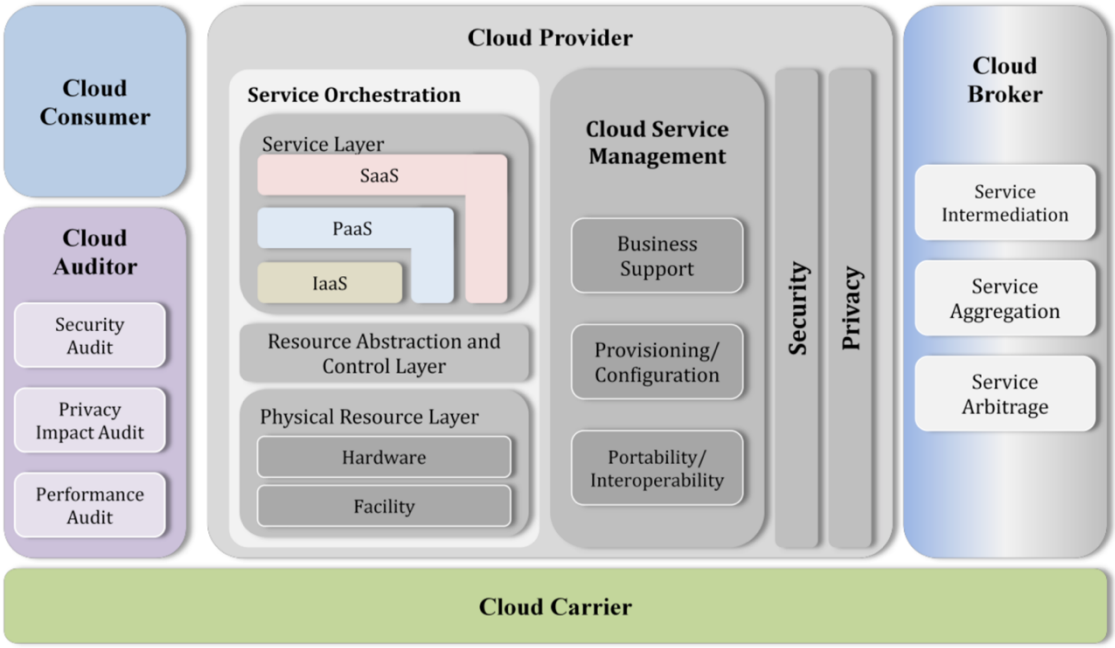


Figure 2-1: NIST Cloud Computing Reference Architecture Model [35]**.**

In terms of roles and interactions, the *Cloud consumer* is a person or an organisation, that can consume any Cloud services offered by a *Cloud provider*, who is responsible for managing and maintaining the Cloud services, or by a *Cloud broker*, who acts as an intermediary between service providers and consumers, and is responsible for ensuring the delivery of Cloud services. The *Cloud auditor* can have the role of collecting, ensuring and verifying essential information in order to evaluate the delivery of Cloud services. Finally, the *Cloud carrier* is responsible for connecting the actors (consumers, brokers and providers) in a Cloud environment [35].

Moving on to the layered design of Cloud Computing architecture, Buyya et al. [36] stated that the Cloud architecture consists of four main levels, namely

system level, core middleware, user-level middleware and user-level Cloud application, as shown in Figure 2-2.

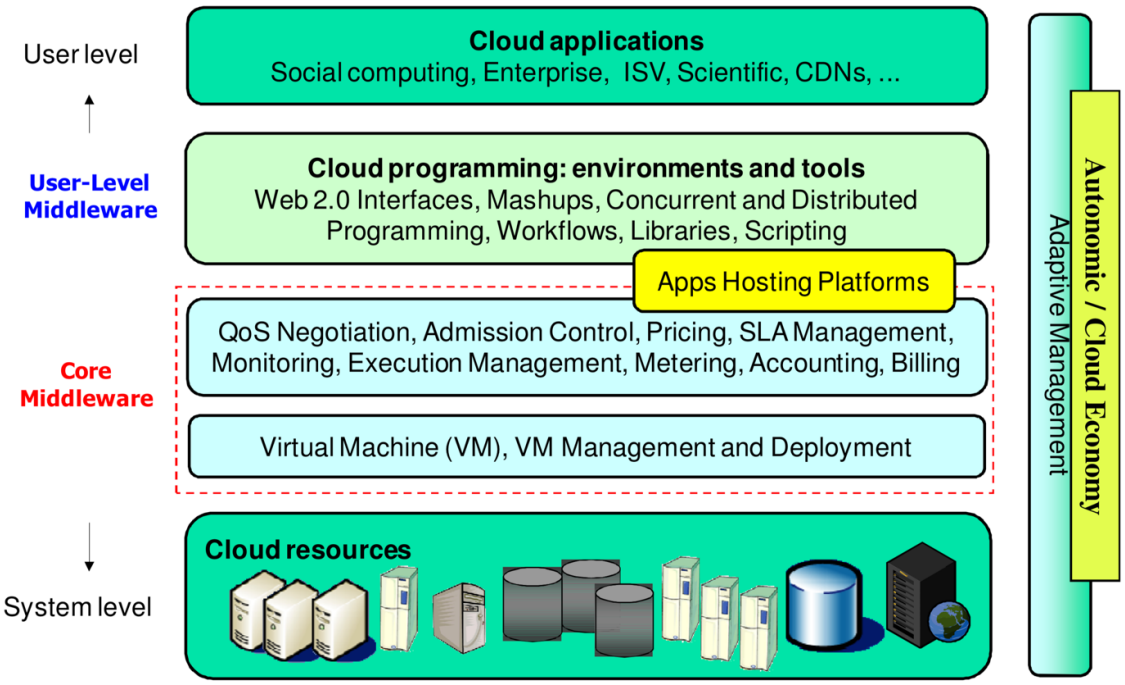


Figure 2-2: Layered Cloud Computing Architecture [36].

Starting from the bottom level, the *system level* is the basis of the Cloud architecture which includes all the physical resources such as servers, routers, switches, network links and storage components. These resources are controlled and managed by the virtualisation services which allow sharing of their capacity among virtual instances [36]. The *core-middleware* level is the platform that provides a run-time environment, enabling to host and control the application services at the user-level middleware. Furthermore, the cloud programming environments and tools are hosted at the *user-level middleware* level in order to support the developers to create and run their applications in Clouds [36]. Finally, the *user-level Cloud application* includes the applications deployed by Cloud providers that can be accessible by the customers [27]. Note that the customers may deploy and run their own applications according to such architecture.

Additionally, Zhang et al. [37] categorised the Cloud Computing architecture into four layers, namely hardware, infrastructure, platform and application layers, as indicated in Figure 2-3.

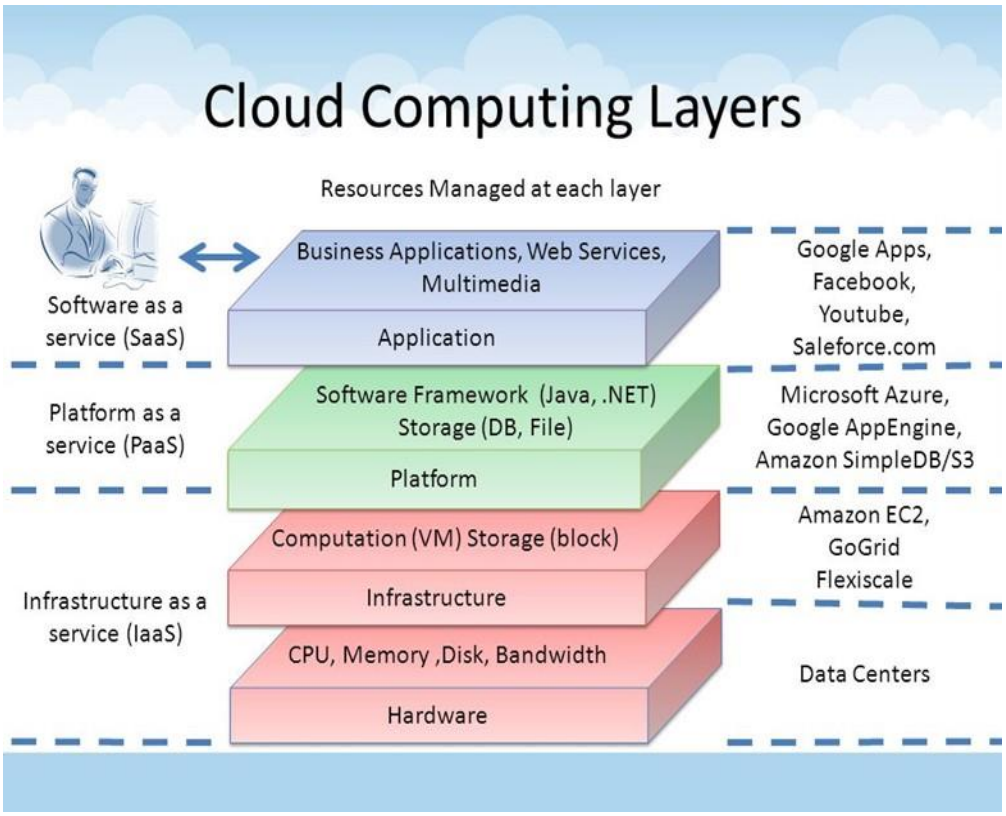


Figure 2-3: Cloud Computing Architecture [37].

At the bottom of this architecture is the *hardware layer* where the Cloud physical resources (e.g., routers, servers, switches and cooling systems) are managed within Cloud data centres [37]. On top of the hardware comes the *infrastructure layer*, which also known as virtualisation layer. The infrastructure layer consists of a pool of virtualised computing resources through the use of virtualisation technologies such as KVM [38], Xen [39] and VMware [40]. On top of the infrastructure layer, the operating systems are included in the *platform layer*, which provides the environment to deploy the applications in virtual instances. Finally, the *application layer* sits on top of the architecture which consists of the actual Cloud applications.

2.2.3 Services Types

With reference to the Cloud architectural layers shown in Figure 2-3, there are three main types of Cloud services, namely, Software as a Service (SaaS) provided at the user level; Platform as a Service (PaaS) provided at the core

middleware; Infrastructure as a Service (IaaS) provided at the system hardware level [37], [27], in addition to other Cloud services such as Everything as a Service (XaaS) [41].

- ***Software as a Service (SaaS)****:* this layer provides applications and software programs, in addition to interfaces for the customers. Over the Internet, customers can access services to utilise applications or software programs and pay fees according to their consumed services, for instance, through *pay-as-you-go* model. Google Apps [42] Google Documents and Google Mail (Gmail) are examples of SaaS service.
- ***Platform as a Service (PaaS)****:* with this type of Cloud service, the customer has the ability to deploy and generate Cloud applications using programming languages, services, libraries and tools supported by Cloud providers. The customer does not control or oversee the cloud infrastructure, including network, storage, servers or operating systems; however, they have control over the deployed applications and often configuration settings for the application's hosting environment. Include examples of the PaaS service are Microsoft Azure Services [43] and Google App Engine [42].
- ***Infrastructure as a Service (IaaS)****:* in this layer, hardware devices and infrastructure are virtualised and offered as a service (e.g., VMs), which also called instances. Several types of virtualisation are supported in this layer on different resources, such as network, computing, hardware and storage. With the IaaS, customers can have access to these resources in order to run their applications; but they do not have the ability to manage the underlying infrastructure that provisions these resources, which is the responsibility of the providers [33]. Amazon Elastic Compute Cloud (EC2) [6] and Rackspace [44] are examples of IaaS service.
- ***Everything as a Service (XaaS)****:* where X is everything that can be described as a new type of Cloud services, such as desktop, network, storage, hardware, security, communication, virtualisation, data and business [41].

2.2.4 Deployment Types

As shown in Figure 2-4, Cloud Computing can be deployed through many models, which can be mainly public, private, hybrid, and community Clouds [33].

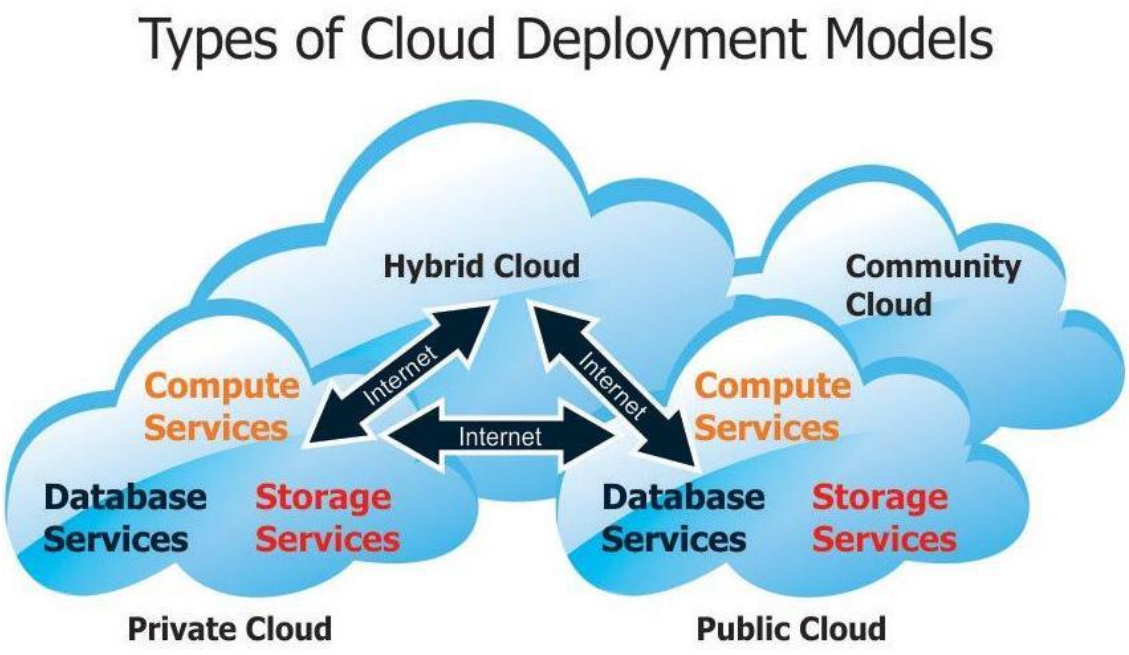


Figure 2-4: Types of Cloud Deployment Models [45].

- ***Public Cloud***: a public Cloud is owned by a service provider offering services and computational resources to organisations and individuals. A public Cloud allows customers access to the cloud through the Internet and the customers only pay for the time period that they utilise the service, (e.g., using a pay-per-use model) [34], [37]. Nevertheless, public Clouds are less secure compared to other Clouds, and all the data and applications on a public Cloud may fall victim to malicious attacks [37].
- ***Private Cloud****:* a private Cloud, which can also be named an internal Cloud or corporate Cloud, is usually hosted and managed by the company itself. Security is improved in a private Cloud as only the company users have access to the provided services. The company owns the cloud infrastructure, which makes it easy to manage applications, resources, maintenance, and upgrades, in addition to providing more control over how applications are deployed [37], [35].

- ***Hybrid Cloud****:* this is a composition of private Cloud and public Cloud. In this type of deployment, a private Cloud is connected to one or more external Cloud services. It allows the company to support its needs in the private Cloud and if extra resources are needed (e.g., at peak time), it can connect to the public Cloud for providing additional computing resources [34], [37], [33].
- ***Community Cloud****:* the community Cloud is a model that is shared between several organisations in order to meet specific requirements that difficult to achieve in a public Cloud (e.g., security requirements, policy, and compliance considerations). In a community Cloud, the infrastructure might be hosted and managed by one or more of the organisations in the community, or by a third-party provider [34], [33].

2.2.5 Virtualisation

Virtualisation is a key component of the Cloud Computing infrastructure and is 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, [46].

According to Hwang et al. [47], virtualisation can be implemented at various operational levels, as given below:

- ***Application level***: it virtualises an application as a virtual machine.
- ***Library support level***: it controls the communication between applications and the rest of a system through Application Programming Interface (API).
- ***Operating system level***: it creates isolated containers on a single physical server and the Operating System (OS) instances to utilise the software and hardware in data centres.
- ***Hardware Abstraction Level (HAL)***: it generates a virtual hardware environment for a virtual machine and manages the underlying hardware through virtualisation.

- ***Instruction Set Architecture (ISA) level***: it emulates a given ISA by the ISA of the host machine.

One of the main advantages of virtualisation is to abstract the Physical Machines (PMs) hardware in order to provide Virtualised Machines (VMs) that can work in isolation and run different applications with different operating systems. By virtualisation, the VMs can be consolidated to minimise the number of active PMs using (e.g., live migration), which would then reduce the power consumption as well as lowering the operational cost, as will be discussed in Section 2.7.1.

Thus, virtualisation adds an essential value to the Cloud infrastructure by increasing the physical resource utilisation, achieving significant energy savings and reducing the operational cost in Cloud environments [48].

2.2.5.1 Virtual Infrastructure Manager

Cloud infrastructure providers use Virtual Infrastructure Manager (VIM) to manage their physical resources in order to provide virtualised resources to meet their customers' service requirements. In order to build, deploy and manage Cloud infrastructures, there are several open source Cloud management platforms available to manage virtualised infrastructures in Clouds. Some examples of the major open source Cloud platforms are OpenNebula [49], OpenStack [50] and CloudStack [51]. The following Table 2-1 summarises some of the features of these VIMs.

Table 2-1: Comparison of Open Source Cloud Platforms.

Functionality	OpenNebula	OpenStack	CloudStack
Cloud Infrastructure	Private, Public and Hybrid Clouds	Private, Public and Hybrid Clouds	Private, Public and Hybrid Clouds
Resource Abstraction	Compute, Storage and Network	Compute, Storage and Network	Compute, Storage and Network
Architecture	Modular (third- party component)	Fragmented into many modules	Monolithic central controller
Installation Difficulty	Easy (process-based package installers)	Difficult (many choices, not fully automation)	Medium (Few parts to install)
Supported Hypervisors	Xen, KVM, VMWare, vCenter	Xen, KVM, VMware, HyperV, vCenter, LXC, vSphere	Xen, KVM, VMWare, HyperV, LXC, vSphere,
Administration	Web UI, CLI	Web UI, CLI	Web UI, CLI
User Management	Yes	Yes	Yes
Live Migration	Yes	Yes	Yes
Load Balancing	Yes	Yes	Yes

Fault-tolerance	VM scheduling, replication	VM scheduling, replication	VM scheduling, replication
High Availability	Yes	Yes	Yes
Security	user authentication	VPNs, firewall, user authentication, others	VPNs, firewall, user management, others
Compatibility	All Amazon Interfaces	Amazon EC2, Amazon S3	Amazon EC2, Amazon S3
Extensibility	Yes	Yes	Yes

OpenNebula, OpenStack and CloudStack have a common role in providing a platform for deploying, managing and provisioning (compute, storage and networking) resources through interfaces such as Web User Interface (Web UI) and Command Line Interface (CLI). However, there are some differences in terms of their architectures based on the configurations, settings and their deployment. For instance, OpenStack has many components to install, which may increase the complexity of installation and configuration as well as the management overhead [52]. In order to avoid this, the OpenStack administrator has to only install the required components to meet the needs of their Cloud deployment. In contrast, OpenNebula does not have such constraints as it provides centralised deployment and has a fine-grained core [52].

In addition to OpenNebula, OpenStack and CloudStack, there are other VIMs available freely or commercially for the deployment and management of Cloud infrastructures such as OpenQRM [53], Eucalyptus [54], Nimbus [55] and others more.

2.2.5.2 Hypervisors

Hypervisors-based virtualisation abstracts the underlying physical hardware to provide isolated instances, called VMs, which can run their own operating system (guest - OS) [56], [57]. These VMs are managed by the hypervisor, which is also referred to the Virtual Machine Monitor or Manager (VMM) to control the number of resources allocated to each VM. The hypervisor sits between the physical hardware and OS, which is also responsible for creating, running, migrating, copying, and deleting the VMs [57]. Further, hypervisors can be implemented in different ways such as full virtualisation when the hypervisor runs on underlying physical OS and hardware virtualisation when the hypervisor runs on underlying physical hardware. Some examples of hypervisors include Kernel-based Virtual Machine (KVM) [38], Xen [39], VMware [40], Microsoft Hyper-V [58] and Virtual Box [59].

2.2.5.3 Containers

Containers-based virtualisation modifies the underlying host OS to provide isolated instances, called *containers*, that can run different applications by sharing the same host OS [57], [60]. Containers provide new ways for faster-running applications, developing, and shipping. It represents a light-weight alternative instance when compared to VM, thus, instead of building one application, developers can build a suite of components, called *micro-services*, which come together over the container [61]. Most of Cloud service providers have moved to *Docker* [62] such as Microsoft, Google and Amazon Web Services to provide the infrastructure that supports the container standard [63]. Containers are better suited to micro-services than VMs, they can start up and shut down more rapidly as well as their resources can be scaled independently. However, containers do not provide full isolation, which may cause security issues. Therefore, hypervisor-based is more appropriate than container-based virtualisation in terms of isolation and security concern. Some examples of containers include Docker [62], Linux Containers (LXC) [64] and Warden Container [65].

2.3 Cloud Computing Applications

Cloud applications should be designed specifically with the support of a Cloud Computing architecture; thus, the applications need to break down into separate components to support the distribution among Cloud resources [66]. Also, the Cloud applications should be designed to support scalability and elasticity, which allow dynamic reservation and release of the Cloud resources to match the changes of the workloads.

2.3.1 Workload Patterns

In Cloud environments, different applications have different resource usage requirements. Therefore, Cloud applications can experience different workload patterns based on the customers' usage behaviours, and these workload patterns consume power differently based on the resources they utilise. As

stated in [67], the cloud workload patterns can be categorised as *static* workload, *periodic* workload, *once-in-a-lifetime* workload, *unpredictable* workload, and *continuously changing* workload.

As depicted in Figure 2-5, a static workload pattern occurs when an application is running continuously with the same and stable resource utilisation over a period of time. Private websites and wikis are examples of such static workload. A periodic workload pattern can be experienced when an application is running with a repeated resource utilisation peaks occurring over time intervals (e.g., seasonal changes). Examples of this type of workload include shopping websites during holiday periods, sporting events (Olympics) and traffic during rush hours.

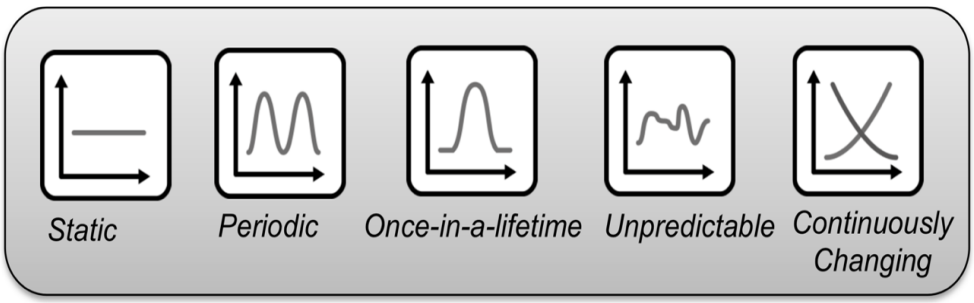


Figure 2-5: Cloud Application Workload Patterns [67]**.**

Furthermore, when an application is running with stable resource utilisation and peak once over time, it is considered once-in-a-lifetime workload pattern. Payroll, billing and backup applications are examples of once-in-a-lifetime tasks or jobs. An unpredicted workload pattern occurs when an application has a random peak (constantly fluctuating) of resource utilisation over time. Unpredictable traffic and forecasting are examples of unpredicted workload. Finally, when the application is running with stable resource utilisation and rapidly decreases or increases over time, it experiences a continuously changing workload pattern [67]. Examples of such type of workload include; social networking (Facebook and Twitter), open-source downloads and Android applications.

As mentioned early, these types of application workload patterns can have a different impact on energy consumption based on the resources they consume. More details on the application workload patterns considered in this thesis are found in Chapter 4, Section 4.2.1.

2.3.2 Benchmarking

Benchmark suites are adopted to evaluate Cloud services to support the configuration and adaptation of applications before they start utilising Cloud resources, such as VMs and containers. Benchmarking aims at defining and reproducing execution conditions for the target system (application, resource, service) to be evaluated [68]. It also provides a set of metrics in order to quantify the relative software and hardware performance, and understand how Cloud application workloads behave as the underlying Cloud resources are stretched and approach full capacity [69].

In this regard, the Standard Performance Evaluation Corporation (SPEC) [70] launched a tool that provides a set of synthetic workloads, which exercises the CPU, memory and disk performance as well as tests the energy efficiency of a system at different load levels. Generally, this benchmark exerts graduated levels of load on a given machine, normally evaluating the energy consumption and performance of server hardware between (idle 0% and fully active 100%) load at 10% graduated load levels [71].

Similarly, a simple benchmarking tool for POSIX systems [72], called *Stress-ng* [73], has been designed as a workload generator. This tool has the capability to simulate a wide range of workload patterns such as static, periodic, continuously changing, and once-in-a-lifetime workload patterns. Further, the *Stress-ng* workload generator is able to simulate both single and multi-threaded applications, as well as test workloads that are resource-bound in many ways, e.g., applications that are both CPU and memory intensive.

More details on the software tool considered in this thesis are found in Chapter 3, Section 3.5.1.

2.4 Pricing Models in Cloud Computing

Cloud service providers offer different types of services to their customers with different pricing models. The strategy of pricing models in Clouds can be categorised as 1) *fixed pricing*: when the price of the services doesn't change (flat fees) and determined by the provider, and 2) *variable pricing*: when the price of the services is dynamically changed based on the market supply and demand [32]. Thus, the price of each Cloud service will be based on the chosen type of pricing model.

The most popular Cloud service providers (e.g., Amazon EC2 [6], Microsoft Azure [7] and Google Cloud [8]) have three common types of pricing models, which are *subscription*, *on-demand* and *auction* pricing models. These pricing models are discussed as follows:

- ***A subscription-based pricing model***: this type of model allows customers to pay a fixed price up-front for a specific period of time, usually monthly or yearly (e.g., reserved instances provided by Microsoft Azure). Typically, customers pay lower prices for long-term commitments due to the fact that this can help Cloud providers to estimate the expenses of their infrastructures [74]. With this type of pricing model, Cloud providers attract more customers' by offering a discount rate and ensuring that their resources will be available at any time they want [32].
- ***A demand-based pricing model***: there are no long-term commitments with this type of pricing strategy, which enables customers to pay service fees on a time-based, usually per hour or second (e.g., pay-as-you-go and on-demand pricing models provided by Google Cloud and Amazon EC2, respectively). Pay-as-you-go model is ideal for businesses that cannot pay up-front or cannot estimate their required computing resources. The price is set according to the size of the instances and their resources. For example, the instances that do not involve Graphics Processing Units (GPUs) or lots of Central Processing Units (CPUs) or Solid-State Drive (SSD) based storage, will automatically be cheaper since they are not used for high performance [32], [75]. Furthermore, a hybrid pricing model is presented by Jelastic plans [76], which is an intermediate model between subscription and on-demand with charged on an hourly basis. In

this model, the customers can set a minimum number of resources to be reserved for an application and get a discount rate accordingly, as well as, it allows the customers to set maximum limits of resources in case if the application demand increases.

- ***An auction-based pricing model***: the idea of the auction pricing model is based on selling the idle time of Cloud services, which enables customers to bid for the services and Cloud providers have the right to accept or reject the offer. For instance, Amazon EC2 Spot instances [77] allow customers to bid on a spare Amazon EC2 computing capacities. Also, customers can view the Spot instance price history for the last 90 days to determine which bid price they should offer [77], [78]. Thus, if the customer's bid exceeds or meets the current bid price, the customer can access the resources. Contrarily, if the customer's bid is overridden, the customer gives the resources back. The prices of the auction-based model compared to subscription and on-demand models are significantly lower. However, if a customer loses a bid, these resources can be taken away, which make it not suitable for businesses [32].

In Cloud environments, the majority of the costs are derived through resource usage, which can be defined as the resource capacity that required to run applications on the cloud infrastructure. However, not all the costs are related to the resource usage of infrastructure, there are further additional costs. For example, the costs that are associated with software licenses, IT support, cooling and maintenance. These costs are difficult to measure or estimate due to the differences between Cloud service providers. Besides, the current pricing models do not provide details of the energy consumed by the offered services. Thus, in order to effectively contribute to the overall business model and offer transparent pricing to the customers, Cloud service providers should consider energy consumption when designing their pricing models [10], [79], [80].

Therefore, only the costs of the cloud infrastructure that can be calculated through resources along with their energy consumption are considered in the scope of this thesis.

2.5 Energy-related Cost Issues in Cloud Computing

Many Cloud service providers such as Amazon [6], Microsoft Azure [7] and Google Cloud [8] have allowed the customers to run their applications in Clouds. They therefore have established cost models in order to charge their customers based on the offered services. Although many cost models in the IaaS are already proposed (e.g., subscription, on-demand, and auction pricing models), there are still inevitable to suffer from wasted payment and resources when using these types of pricing models [81], [10], [82]. In fact, cost modelling is a critical component of the Cloud Computing paradigm since it directly affects providers' revenue and customers' payment [81], [82]. Thus, designing an appropriate and precise cost model which can make both providers and customers satisfied is considered as a vital concern in a Cloud environment.

Furthermore, Cloud data centres continue to consume huge amounts of energy and have a major impact on environmental and operational costs caused by this high energy consumption [83]. With the increasing electricity costs for Cloud data centres, energy consumption has become one of the major operational cost issues for Cloud providers to maintain [84]. In 2013, Cloud Computing consumed about 684 billion Kilowatt-Hour (kWh) of electricity [85], while the increase in energy consumption is estimated to be around 60% or even more by 2020 [85]. Yet, most of Cloud service providers charge their customers for the offered services on a time-based without considering the actual cost of energy consumption [23]. Due to the economic impact of Cloud data centres' energy consumption, Cloud providers should consider the actual cost of energy consumption when designing their cost models for the offered services [10]. In addition to that, Cloud customers cannot affect or know in any way the amount of energy that they consume for running the cloud services. Consequently, it is necessary to make them aware of their energy usage, which may help to change their behaviour accordingly, for example, by shutting down/consolidating VMs and running applications which are energy efficient.

In the following subsections, some of the research conducted on Cloud cost models to reduce the cost and energy consumption in the IaaS Cloud environment will be discussed.

2.5.1 Cost Models

Cloud cost modelling is a challenging issue as the increasing number of business are moving their computation workloads to Clouds. Although many public Cloud providers are already used the (*pay-as-you-use*) model to charge their customers for the offered services, the customers still usually pay more than what they are actually used [86]. Therefore, the work by Belli et al. [87] explored the area of cost models, that allows the customers to optimise their choice of IaaS Cloud providers in terms of the offered price. They presented a Cost-Optimised Cloud Application Placement Tool (COCA-PT) based on a Resource Consumption Model (RCM). The main goal of their work is to optimise the placement of customers applications based on the price offered by different Cloud providers. However, as mentioned by the authors, the proposed tool is not completed yet and needs a further extension. Moreover, their cost model does not take into account the power consumption consumed by the running applications.

Jin et al. [81] designed a fine-grained fair pricing model to improve the resource utilisation and reduce partial usage waste problem. They investigated the optimisation of the trade-off between the proposed model and various overheads (e.g., VM maintenance and billing cycle). The model is evaluated using two large-scale production traces (Grid Workload Archive and Google Data Centre) and the experimental result show that the proposed model can significantly improve social welfare (e.g., increasing provider revenues and reducing customers costs). Although the authors have been focused on the design of precise pricing model that can satisfy both customers and providers, their approach has not shown the impact of the power consumption of the used resources on Cloud pricing models.

Moreover, Berndt and Maier [23] presented a hybrid IaaS pricing model to address an issue when Cloud providers practice of overbooking and double selling capacity in order to retain profitability, which would affect performance and Cloud adoption. To clarify, this pricing model charges based on a flat rate part that guarantees a certain performance to the customers and on a flexible part that charges for the resource usage exceeding the flat rate portion. Their approach only requires measurement of performance in one side and

measurement of resource usage on the other side, as stated in their work [23]. Yet, their approach is still limited in the essence that it does not consider the actual cost of energy consumption.

Mao and Humphrey [17] presented a cost-aware auto-scaling mechanism for scheduling tasks in Clouds, which called Scaling-Consolidation-Scheduling (SCS). The auto-scaling mechanism takes into consideration the instantiation time that every VM needs to be running, then the Earliest Deadline First (EDF) algorithm is used to schedule tasks on each VM. They primarily focus on minimising the cost of the VMs and satisfying their performance requirement based on tasks deadline constraints. This is achieved by forcing the tasks to run on the same VM in order to improve performance and save the data transfer cost. They compare the proposed SCS approach with two cost-based approaches, and the results demonstrate that their approach achieved cost-savings of 9.8% - 40.4% along with improved utilisation over other approaches. However, this approach only ensures a reduction in the cost of each VM and does not take into account the trade-off between performance and power consumption of the selected VMs.

Further, a cost-aware super professional executor (Suprex) with auto-scaling mechanism is proposed by Aslanpour et al. in [88]. This approach aims to provide an executor with the capability to isolate the overloaded VM until the billing period is completed, which leads to overcome the challenge of postponed VM start-up and maximise the cost efficiency. The results show that the Suprex executor can reduce the cost of VM by 7%, but in some cases this executor leads to lower resource utilisation.

Chard et al. [89] proposed an approach for cost-aware elastics resource provisioning for scientific workloads. This approach monitors a job submission queue and provisions VMs based on pre-defined policies. The authors investigate the impact of workload execution on the total cost of Cloud services by using dynamic pricing models (e.g., Spot and On-Demand instances) provided by Amazon Web Services (AWS) [77], based on different availability zones. They evaluate their approach under realistic conditions based on workload traces through simulation. However, their investigation does not consider the impact of energy consumption on AWS pricing models.

2.5.2 Cost and Energy Consumption Models

With the expansion of Cloud Computing, optimising the energy efficiency of the Cloud paradigm at all different layers is considered significantly important, as highlighted by Djemame et al. [90], [91]. The authors have proposed a Cloud architecture that enables energy awareness at all layers of the Cloud stack and through the Cloud application life-cycle. This architecture is a complete energy efficient solution, capable of self-adaption and aware of the impacts on other quality characteristics such as cost and performance of the applications. An example of a cost model for Infrastructure as a Service (IaaS) providers to align with the energy consumption cost is introduced by Hinz et al. in [10]. They proposed a cost model called Proportional-Shared Virtual Energy (PSVE), which investigates the relationship between energy consumption and VMs workload in a Cloud environment. The PSVE model considers the cost of heterogeneous VMs as well as their energy consumption, which is based on the number of allocated virtual CPU to each VM. Also, it consists of two main elements: 1) a cost associated with VMs resources (e.g., CPUs and networks) along with their power consumption, and 2) a shared cost associated with the hypervisor, relatively distributed among VMs. Nevertheless, their model does not consider the actual utilisation of the virtual CPUs, only considers the number of allocated virtual CPU to each VM, thus their cost model may not be an accurate, as stated by [79], [80]. In this context, the current cost models offered by Cloud service providers (e.g., Amazon EC2 [6]) only consider the number of allocated resources to each VM based on the time of usage, and do not consider the utilisation ratio of these resources (actual usage). To illustrate that, let's consider two VMs (VM1 and VM2) allocated on the same host and have the same number of virtual CPUs. VM1 and VM2 used 10% and 90% of the CPU utilisation, respectively. These ratios of CPUs utilisation have different impacts on energy consumption [92], but both are usually charged the same price, regardless if a VM is using 10% or 90% of its CPU. Consequently, the authors in [79], [80] highlighted the need of Cloud service providers to offer cost models that fairly charge their customers based on the actual resource usage with consideration of their energy consumption.

Wang et al. [75] argued the importance of having precise cost models for adopting Cloud Computing. Through their investigation, they found that different system configurations have a significant impact on energy consumption and thus the total cost of Cloud services. Consequently, Yousefipour et al. [93] proposed an energy and cost-aware VM consolidation model that aims to minimise the number of active PMs in order to reduce power consumption and cost of heterogeneous Cloud data centres. The consolidation process is nearly optimised based on the trade-off between power consumption and cost using a mixed-integer non-linear programming model and a genetic algorithm. The results show that the proposed model is capable of reducing the power consumption and costs when compared to the First Fit (FF), First Fit Decreasing (FFD), and Permutation Pack (PP) algorithms. However, they assume the power consumption is increasing linearly for all PMs, which is not usually the case in the heterogeneous Cloud environment, as stated in [94]. Also, this work does not consider the energy consumption overhead incurred by VMs consolidation. Similarly, in [95] authors proposed a cost and energy efficient scheduling algorithm based on Particle Swarm Optimization (PSO). This algorithm aims to optimise execution cost and energy consumption of Cloud data centres, considering deadline constraint and time. The proposed algorithm is evaluated using CloudSim [27] based on independent tasks scheduling and compared with honey bee and min-min algorithms. Nevertheless, this work lacks to consider other Quality of Services (QoS) parameters such as application performance variations, load balancing, availability and SLA violation.

Further, Jung et al. [12] introduced Mistral, a holistic framework that balances the power consumption, application performance and the transient power/performance costs incurred by the adaptation decisions of the framework. Their approach investigates the problem of dynamic consolidation of homogeneous VMs and focuses on improving the power consumption of the physical host. However, this framework does not optimise the trade-off between all mentioned objectives (power consumption, application performance and costs), only two of these objectives are considered to be optimised at the same time.

2.5.3 Overall Discussion

Cost modelling is an important component of the Cloud Computing paradigm since it directly affects providers' revenue and customers' payment [81], [82]. The main aim for Cloud providers is to achieve maximum revenue and for Cloud customers to achieve the highest service performance at a reasonable price.

Current cost models used by Cloud service providers (e.g., Amazon EC2 [6] and Microsoft Azure [7]) are based only on the usage of the virtualised resources such as CPU, memory, and disk, and do not consider the variable cost of energy consumed by these resources. With the increasing cost of electricity, Cloud providers consider energy consumption as one of the most important operational cost factors to be managed within their infrastructures [1]–[3]. Consequently, modelling a new cost mechanism for Cloud services that takes into account the actual energy costs has attracted the attention of many researchers [1]–[3].

Although many public Cloud providers are already using the *(pay-as-you-go*) model to charge their customers for the offered services, the customers still usually pay more than what they are actually use [81], [82], [86]. Therefore, designing an appropriate and accurate cost model which can make both providers and customers satisfied is considered as a vital concern in a Cloud environment.

In order to properly alleviate the operational cost, Cloud service providers can be assisted with cost and energy awareness to enhance their decisions and efficiently manage Cloud resources. Section 2.5 has reviewed the related work on modelling the cost as well as the energy consumption in Cloud environments. As discussed in Section 2.5.1, the work presented in [87], [81], [23], [17], [89] aimed to improve the cost efficiency in Cloud environments in order to meet the performance requirements, customers' demands and efficient resource utilisation, but not considering the energy consumption of the resources. In Section 2.5.2, the work presented in [93], [12] considered the energy consumption in their models, but their focus is only at the physical level in order to consolidate the VMs and minimise the number of active hosts. Only the work presented in [10] considered the energy consumption at both physical and virtual levels, though this is still limited as their model only consider the number of

allocated virtual CPU to each VM. Thus, there is a clear need to consider energy consumption at VMs level, taking into account the actual utilisation of the virtual CPUs in order to obtain a precise cost model.

The following Table 2-2 provides a comparison summary of the closely related work on modelling cost and energy consumption for VMs in a Cloud environment.

Table 2-2: Summary of Existing Cost and Energy Models.

Criteria / by	Cost Model based on VMs Resource Utilisation Consideration	Actual Power Consumption Consideration	
		PMs level	VMs level
Belli et al. [87]	Homogeneous VMs only.	Not considered.	Not considered.
Jin et al. [81]	Homogeneous VMs only.	Not considered.	Not considered.
Berndt and Maier [23]	Homogeneous VMs only.	Not considered.	Not considered.
Mao and Humphrey [17]	Homogeneous and heterogeneous VMs.	Not considered.	Not considered.
Chard et al. [89]	Homogeneous and heterogeneous VMs.	Not considered.	Not considered.
Yousefipour et al. [93]	Homogeneous and heterogeneous VMs.	Homogeneous PMs only.	Not considered.
Jung et al. [12]	Homogeneous VMs only.	Homogeneous PMs only.	Not considered.
Hinz et al. [10]	Homogeneous and heterogeneous VMs.	Homogeneous PMs only.	Homogeneous and heterogeneous VMs, but only based on the number of allocated virtual CPUs to each VM.

2.6 Prediction Models in Cloud Computing

Having discussed the existing work on modelling the cost and energy consumption of Cloud services in Sections 2.5.1 and 2.5.2, this section discusses the work on predicting the workload, energy consumption and estimating the total cost of the VMs during the service operation.

Providing prediction information of the Cloud services ahead of their operation or at the run-time can be very beneficial for the service providers, as they need to carefully predict their business growths and efficiently manage the Cloud resources.

To optimise the use of Cloud services, *proactive* mechanisms can be applied to improve resource utilisation and reduce energy-related costs, while

maintaining service performance requirements. However, such mechanisms need to be supported with performance and energy awareness not only at the physical machine (PM) level but also at virtual machine (VM) level in order to make enhanced cost decisions. Moreover, estimating the future cost of Cloud services can help Cloud service providers offer suitable services that meet their customers' requirements.

2.6.1 Workload Prediction

In terms of workload prediction, a number of methods are used in order to predict the workload in Cloud environments. For example, an evaluation of commercial Cloud services offered by major service providers is provided in [96], where a Cloud monitoring tool is used to measure the service performance of a month period for 20 Cloud providers. According to the workload data collected from different Cloud providers, they applied the Auto-Regressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ETS) to predict the future behaviour of service performance. This prediction helps Cloud customers and service brokers to select Cloud services according to their requirements. The overall performance prediction results show that ARIMA performs better than ETS for predicting service performance. Also, a predictive elastic resource scaling scheme (PRESS) for Cloud systems is presented in [97]. The approach uses a short-term pattern matching and state-driven approach (Markov chain) to predict the PMs and VMs workloads. This approach is implemented on top of Xen [39], using RUBiS [98] and an application load traces from Google. In their work, only the workload as a standalone application is predicted.

Moreover, Huang et al. [99] proposed an elastic resource allocation mechanism for a Cloud system, namely Prediction-based Dynamic Resource Scheduling (PDRS). The PDRS is employed to predict the VMs workload fluctuations using the ARIMA model based on the historical workload data. Based on the predictor, they developed dynamic resource allocation algorithms along with VMs live migration in order to reduce the number of active PMs. The results show that this approach is able to realise adaptive resource allocation with an acceptable effect on SLAs and migration overhead. Though this approach is focused on predicting the workload in order to perform the VMs

allocation and live migration, without considering the energy overhead due to the migrations of the VMs.

Farahnakian et al. [100] introduced a predictive VM consolidation approach, called Utilization Prediction-aware VM Consolidation (UP-VMC). The UP-VMC aims to optimise three objectives include the number of SLA violations, the number of VM migrations and energy consumption. It considers the current and future PMs and VMs resource utilisation in order to migrate VMs into the least number of active PMs, and then switch the idle PM to the sleep mode in order to minimise the energy cost. The future resource utilisation (CPU and memory) is predicted using two regression-based prediction models (Linear and K-Nearest Neighbour). The obtained results using Google cluster and PlanetLab workload traces show that the UP-VMC can reduce SLA violations, energy consumption and the number of migrations. However, the experiments conducted on a simulation-based have focused on predicting the PMs and VMs resource utilisation and do not consider the prediction of PMs and VMs energy consumption.

Zhang et al. [20] presented a proactive virtual resource management framework, called (PRMRAP), which predicts the amount of resource needed to cope with unexpected workload changes. This approach uses the ARIMA model based on the historical workload data in order to predict the VMs workload changes and the number of resources needed. In this framework, they consider both vertical and horizontal scaling of the VMs, which can reduce time latency for handling the workload changes in a cost-efficient manner. Likewise, Fang et al. [101] presented a novel Resource Prediction and Provisioning Scheme (RPPS), which predicts the workload demands and dynamically adjusts resource provision for Cloud applications. This approach takes advantage of the ARIMA model which has high prediction accuracy in order to handle the resource provisioning in a short period of time. They implemented the RPPS model on top of Xen [39] and KVM [38] virtualisation platforms, and conducted the experiments in a real Cloud data centre. The results show that this approach has high prediction accuracy of about 90% and able to scale Cloud resources under different situations (e.g., peak and low phases). Further, Yang et al. [102], [18] used a Linear Regression Model (LRM) to predict the VMs workload for the next time interval. Based on the predicted workload, an auto-scaling mechanism is

proposed to scale virtual resources which combines the real-time scaling and the pre-scaling in order to handle the workload demands. They used the knowledge from workload prediction to select the number of resources needed for scaling, considering both horizontal and vertical scaling. According to the experiment results, this approach is able to predict the VMs workload while lowering the scaling costs and Service Level Agreement (SLA) violations. However, all these approaches do not consider or predict VMs energy consumption when performing dynamic resource provisioning (scaling decisions).

Also, it is worth mentioning that the workload prediction modelling requires a quantitative evaluation and statistical analysis in relation to the characteristics of the workloads in terms of their length, pattern, and resource consumption. Thus, modelling the relationships between these different workload characteristics is important in order to achieve accurate and reliable prediction results.

2.6.2 Energy Prediction

There are many ongoing research projects focusing on the prediction of energy consumption based on resource utilisation. For example, Bircher and John [103] proposed an approach to estimate the power consumption of a complete system using microprocessor performance counters. They developed power models for subsystems (e.g., CPU, memory, disk, and network) on two platforms (server and desktop). Also, synthetic workloads were generated in order to control the utilisation of the subsystems. They performed a correlation analysis between the performance counters and the power consumption using linear and polynomial regression techniques. The average error of their models was 14.1% for the memory controller and less than 9% for each subsystem. Similarly, McCullough et al. [104] have evaluated the competence of existing predictive power models based on their accuracy using hardware performance counter for modern hardware architectures. A number of linear and non-linear regression models are compared. For the linear regression models, the results show that these models provide a reliable accuracy with low computational complexity for a single-core scenario. In contrast, the non-linear models provided better accuracy with the multi-core scenario, although they incur a higher computational complexity.

However, these approaches are performed on non-virtualised environments and thus do not consider or support the power consumption of the virtual resources.

Smith et al. [105] proposed a power monitoring tool for software-based, called CloudMonitor. The authors argued that such a tool can be used in order to create energy-efficient applications as well as design energy-based cost models. The results show that the power monitoring tool is able to estimate PM power consumption for different applications as long as the physical hardware has the same configuration. However, this tool does not support the heterogeneity of the PMs as well as the estimation of VMs power consumption.

Kistowski et al. [106] introduced a model for predicting the power consumption of physical hosts at run-time. This approach makes use of run-time monitoring data to train the model and then predict the power consumption based on load intensity and performance counters. The authors claimed that this approach can be used with any performance model to optimise the energy efficiency of distributed systems. They evaluated the model using two different web applications deployed in a heterogeneous environment. The results show that this approach can predict the power consumption of a system with an error of 2.21%. Yet, their approach only considers the prediction of the power consumption at the PMs level and does not consider the prediction at the VMs level.

Further, Makaratzis et al. [107] conducted a survey study on energy modelling in Cloud simulations. They focused on the energy models that have been proposed for the prediction of the energy consumption of Cloud data centres. The most popular Cloud simulation frameworks were considered in this survey: CloudSched, CloudSim, DCSim, GDCSim, GreenCloud and iCanCloud. Hence, the experiments were conducted in order to compare these different simulations with their energy models, and the results show that the same tendency prevails for the energy models in all Cloud simulation frameworks. However, these simulations along with their energy models do not consider the impact of heterogeneous VMs on the energy consumption in Cloud data centres.

Li et al. [48] have built an online power metering model that estimates the power consumption for the PMs and VMs in a Cloud environment. The power modelling is performed using a linear regression technique based on the impact

of the CPU, memory and disk. The implementation of the model shows that it can achieve an average estimation accuracy of more than 96% with low runtime overhead. Nevertheless, they assumed that all the PMs and VMs are homogeneous, which is very rarely used in Cloud environments.

Moreover, Farahnakian et al. [108] introduced a load prediction method, called a Linear Regression-based CPU Utilisation Prediction (LiRCUP). This method is used to predict the short-time future CPU utilisation of the overloaded and underloaded PMs based on historical data of each PM. Based on this prediction, some VMs are migrated to other hosts in order to avoid SLA violations and reduce energy costs. In order to evaluate this work, the authors implemented the proposed method in the CloudSim and the results show that the proposed method can reduce the energy cost and SLA violation rate. However, this work is focused on predicting the workload and then the energy consumption only at the host level and not considering the workload and energy prediction at the VM level.

Subirats and Guitart [109] proposed a VM placement algorithm, which is aimed to take the appropriate decisions (e.g., VM replication, migration, cancellation). Generally, mathematical modelling is used to design a CPU utilisation predictor in order to predict the energy consumption for different workload types at PMs and VMs levels. Their proposed predictor consists of four prediction models, namely linear regression, moving average, single and double exponential smoothing in order to predict CPU utilisation and power consumption of a given VM. Although this work only considers a linear relationship between the CPU utilisation and the power consumption, other non-linear relationships such as polynomial and exponential could be considered in order to increase the prediction accuracy.

2.6.3 Cost Estimation

Estimating the cost of resource provisioning is essential to automatically cope with workload demands. Therefore, Jiang et al. [19] presented an online temporal data mining system, called A Self-Adaptive Prediction (ASAP), which is used to predict the VM demands, and provision resources accordingly. The authors also proposed a Cloud Prediction Cost which is used to measure the performance of

several prediction models based on historical time series data. The experiments results show that the ASAP is capable to decrease the resource provisioning time of all VMs. Another approach for an efficient auto-scaling is proposed in [110]. They used a second order Auto-Regressive Moving Average (ARMA) model in order to predict the VMs workload and cost for the next time interval based on historical workload data. This look-ahead approach enables early auto-scaling detection, which allows the new VMs to boot (horizontal scaling) before workload increases. The model aims to minimise resource usage and satisfy QoS requirements, while keeping operational costs low. However, these two approaches only consider workload prediction for dynamic resource provisioning, and do not consider the energy consumption which would influence the overall cost of the scaling decisions.

Sharma et al. [111] proposed a cost-aware resource provisioning framework for Cloud applications, called Kingfisher. It aims to optimise the cost of resource provisioning and reconfiguring using Integer Linear Program (ILP) formulation. Kingfisher exploited both scaling and migration mechanisms to dynamically select appropriate decisions that optimise the cost incurred by customers. In their work, the ARIMA model is employed to estimate the workload in order to capture future workload trends. They implemented the Kingfisher framework using the OpenNebula Cloud platform, and the results demonstrate that the Kingfisher has the ability to select the lowest cost of resource provisioning and reconfiguring to meet an application’s requirements. Nevertheless, their approach does not consider the energy consumption overhead when performing the migration and scaling decisions.

Furthermore, Liu et al. [112] designed performance and energy models to estimate VM migration cost based on theoretical analysis and empirical studies on the Xen platform. The theoretical analysis and empirical studies show that the migration-related parameters like VM memory size, memory dirtying rate and network speed are the major factors impacting migration performance in terms of migration time, migration downtime and the total volume of network traffic. Also, they designed a linear regression model and a theoretical model to estimate the energy consumption of the networks during VM migration based on their performance model. The experimental results demonstrate that the proposed models are able to estimate VM migration cost with an estimation

accuracy of about 90% based on performance and energy metrics. However, this work does not consider the heterogeneity of the PMs or the VMs when designing their models.

2.6.4 Overall Discussion

Cloud service providers can take advantage of prediction models to enhance the efficiency of managing Cloud resources. With the unexpected workload demands, Cloud service providers should strike a balance between their operating costs, energy consumption and satisfying QoS objectives. Consequently, modelling a proactive mechanism can be beneficial to improve resource utilisation and reduce energy-related costs, while maintaining service performance requirements.

Section 2.6 has reviewed the related work on predicting the workload and energy consumption as well as estimating the total cost of the VMs during the service operation. As discussed in Section 2.6.1, the work presented in [18], [20], [97], [99]–[102] aimed to predict the workload in order to improve resource utilisation in Cloud environments, but without considering the energy consumption of the predicted workloads.

In Section 2.6.2, the work presented in [105], [106], [108] considered the prediction of energy consumption in their models, but these approaches only consider the prediction of the power consumption at PMs level and do not consider the prediction at VMs level. Only the work presented in [48], [109] considered the prediction of energy consumption at both physical and virtual levels. Though there are still limited as the model in [48] assumed that all the PMs and VMs are homogeneous, whereas, the model in [109] only considers a linear relationship between the CPU utilisation and the energy consumption in order to predict the power at the VMs level.

The work presented in [19], [110], [111] considered the prediction of workload and the estimation of cost for the VMs, but do not consider the energy consumption which would influence the overall cost estimation of Cloud services, as discussed in Section 2.6.3. The only work that considered the prediction of workload and energy consumption as well as the estimation of cost, is presented

in [112]. However, this work does not consider the heterogeneity of the PMs or the VMs when designing their models.

Thus, there is still a need for predictive modelling that takes into account the workload, energy consumption and cost not only at the PMs level, but also at the VMs level considering their heterogeneity, in order to make enhanced cost decisions and efficiently manage Cloud resources.

The following Table 2-3 provides a comparison summary of the closely related work on prediction models that consider workload, energy consumption and cost for VMs in a Cloud environment.

Table 2-3: Summary of Prediction Models.

Criteria / by	Workload Prediction Consideration		Energy Prediction Consideration		Cost Estimation Consideration
	PMs level	VMs level	PMs level	VMs level	
Gong et al. [97], Huang et al. [99]	Homogeneous PMs only.	Homogeneous VMs only.	Not considered.	Not considered.	Not considered.
Farahnakian et al. [100]	Homogeneous and heterogeneous PMs.	Homogeneous and heterogeneous VMs.	Not considered.	Not considered.	Not considered.
Zhang et al. [20]	Not considered.	Heterogeneous VMs.	Not considered.	Not considered.	Not considered.
Fang et al. [101]	Homogeneous PMs only.	Not considered.	Not considered.	Not considered.	Not considered.
Yang et al. [102], [18]	Not considered.	Heterogeneous VMs.	Not considered.	Not considered.	Not considered.
Smith et al. [105]	Not considered.	Not considered.	Homogeneous PMs only.	Not considered.	Not considered.
Kistowski et al. [106]	Not considered.	Not considered.	Heterogeneous PMs.	Not considered.	Not considered.
Li et al. [48]	Not considered.	Not considered.	Homogeneous PMs only.	Homogeneous VMs only.	Not considered.
Farahnakian et al. [108]	Heterogeneous PMs.	Not considered.	Heterogeneous PMs.	Not considered.	Not considered.
Subirats and Guitart [109]	Heterogeneous PMs.	Homogeneous VMs only.	Heterogeneous PMs.	Homogeneous VMs only.	Not considered.
Jiang et al. [19]	Heterogeneous PMs.	Heterogeneous VMs.	Not considered.	Not considered.	Based on the resource usage.
Roy et al. [110]	Not considered.	Homogeneous VMs only.	Not considered.	Not considered.	Based on the resource usage.
Sharma et al. [111]	Heterogeneous PMs.	Homogeneous VMs only.	Not considered.	Not considered.	Based on the resource usage.
Liu et al. [112]	Homogeneous PMs only.	Homogeneous VMs only.	Homogeneous PMs only.	Homogeneous VMs only.	Based on the resource usage and power consumption cost for homogeneous PMs and VMs.

2.7 Dynamic Resource Management in Cloud Computing

Resource management is one of the most important problems in Cloud infrastructures, which can be expressed as a multi-objective problem since there are several conflicting objectives (e.g., maintain the performance, reduce energy and costs) that need to be optimised [113], [15]. Therefore, Cloud service providers have applied dynamic resource management through VMs' consolidation and resource provisioning techniques in order to meet the performance requirements of applications, while minimising the operation costs and energy consumptions in Cloud data centres.

In the following subsections, VMs' consolidation and resource provisioning along with their related works will be discussed.

2.7.1 VM Consolidation

One of the benefits of virtualisation is the VMs' consolidation strategy, which allows Cloud service providers to migrate and reallocate the VMs from one host to another in order to increase resource utilisation and reduce energy costs in Cloud data centres [114]. Hence, VMs' consolidation through live migration has a major impact on energy efficiency by gathering several VMs into the minimum number of hosts and switching the idle hosts to a power-saving mode. However, VM consolidation is not a trivial task in case of unpredicted increases in demand, as it can result in generates unnecessary migrations, violations of the SLA and increases the operation cost due to the migration processes [115]. Therefore, dynamic VMs consolidation requires an estimate of the workload demand in order to handle the fluctuating demands of Cloud customers, efficiently manage Cloud resources and avoid unnecessary migrations [116].

VM live migration acts as a backbone of the VM consolidation process, which can be defined as the capability of transferring a complete state of the VM (including CPU states, memory pages, storage and network connections) from the source host to the destination host, without any interruption in the service or application [117], [118]. There are two types of VM migration, which are currently used in Cloud data centres, namely, *post-copy* and *pre-copy* migration.

- **Post-copy**: transfers a VM's memory contents after its processor state has been sent to the destination host. However, this method can take a long migration time, which consumes the resources on both source and destination hosts due to the residual dependency. Also, it has some downtime initially, which makes the VM's service unavailable for a certain time period [119].
- **Pre-copy**: first copies the memory state to the destination, through iterative phases, after which its processor state is transferred to the destination. In this way, the VM can be migrated from one host to another with a close to zero downtime [120].

Live migration efficiency of multiple VMs has been investigated in various research studies. For instance, Ye et al. [117] presented a live migration framework of multiple VMs based on different resource reservation mechanisms. This framework aims to improve migration efficiency by using parallel migration and workload-aware migration strategies. Experimental results show that the performance overheads of the live migration process are affected by workload types, memory size and the number of CPUs. Thus, parallel migration and workload-aware migration strategies can efficiently improve the performance of migrated VMs. However, the performance overhead incurred by concurrent VM migrations may increase the migration interference on the destination host.

Zhao et al. [121] presented a VM placement method based on VM service performance, which aims to address VMs performance degradation issue when placing the VMs. This method takes the application-aware resource consumption characteristic into consideration to place the VMs on appropriate PMs in order to guarantee the VM performances and ensure customers' Quality of Experience (QoE). The proposed method is evaluated in a real Cloud platform (OpenStack) using video streaming applications. The results show that the proposed method can minimise PM performance degradation and guarantee the VM performance compared to other methods. However, their approach only focuses on the resource consumption characteristic when performing VMs placement and does it not take the power consumption of the PMs and VMs into account.

Moreover, Ferreto et al. [122] proposed an approach called dynamic consolidation with migration control, which aims to reduce the number of VM

migrations and the number of active hosts using linear programming formulation. This approach gives a higher priority to migrate VMs with variable workload instead of the VMs with a stable workload in order to reduce the number of migrations and required hosts with a minimal SLA violation. They compared the proposed approach with static and dynamic consolidation approaches using TU-Berlin and Google data centre workloads. The evaluation results demonstrate that the suggested approach performs well in terms of the number of PMs used and VMs migrated. However, this approach does not take into account VMs power consumption and migration costs when consolidating the VMs.

Farahnakian et al. [118] presented a modified approach of Best Fit Decreasing (BFD) algorithm, named a Utilization Prediction-aware Best Fit Decreasing (UP-BFD) algorithm. This approach employed a utilisation prediction model to eliminate unnecessary VM migrations and reduce SLA violations using K-Nearest Neighbor Regression (K-NNR) model. The prediction model is trained by generating historical data based on different types of workloads developed in the CloudSim. This approach also considers both the current and future utilisation of resources in order to perform VM consolidation based on the hosts CPU and memory utilisation thresholds. Although this work focuses on reducing PMs energy consumption, the number of VM migrations and SLA violations, they do not consider the impact of energy consumption that occurs by VMs live migration decisions in their approach.

Further, Beloglazov and Buyya [123] addressed the problem of VMs consolidations under QoS constraints in Cloud data centres. They employed the Markov chain model and the control algorithm to detect the overloaded hosts and then migrate some VMs in order to achieve a specified QoS goal. This dynamic VMs consolidation aims to improve the PMs resource utilisation (particularly CPU utilisation) for stationary workloads, which also can be applied for non-stationary workloads using the Multisize Sliding Window workload estimation technique. Simulation results using workload traces on PlanetLab servers demonstrate that the introduced method outperforms the benchmark methods while meeting the QoS goal. However, this method focused on improving the performance of Cloud applications by reducing the number of overloaded hosts, but without explicitly considering energy and cost of VMs migrations, as a part of VMs consolidation decision criterion.

Xu et al. [124] proposed a lightweight interference-aware VM live migration strategy, called iAware. It focuses on the performance of VMs during and after live migration, considering the interference of the migration process on both source and destination PMs. The iAware jointly estimates, analyses and minimises both the migration time and co-location interference among VM's based on a multi-resource demand and supply estimation model. The experiments are conducted in a real Cloud environment with different workloads using a Xen hypervisor cluster platform. The results are compared with traditional interference-unaware algorithms and show that the iAware can estimate VM performance interference during live migration and meet the SLA requirements. However, their work does not consider the energy consumption overhead of VMs migrations.

Beloglazov and Buyya [125] presented an energy efficient resource management policy for Cloud data centres. The proposed method mainly focuses on dynamic re-allocation of VMs using live migration in order to minimise the energy consumption, while maintaining the QoS requirements. They evaluated the proposed method using a CloudSim and the results show a reduction of energy consumption in a Cloud data centre. However, the proposed method does not show the effectiveness of the heterogeneity of the PMs in terms of energy efficient when performing the live migration of the VMs.

Furthermore, Beloglazov et al. [126] presented an energy-aware VM consolidation policies to optimise the resources utilisation and energy efficiency in a Cloud data centre. In this approach, the VMs are migrated from one host to another in order to increase the overall servers' utilisation and reduce infrastructure costs (energy costs) by switching off the idle hosts. Thus, upper and lower CPU utilisation thresholds for each host are set along with several VM selection policies, in order to identify from which host the selected VMs should be migrated. The experiment results conducted in the CloudSim show that this approach leads to an improvement of energy efficiency in Cloud data centres. Likewise, Farahnakian et al. [113] proposed a Self-Adaptive Resource Management System (SARMS) for efficient resource management in Cloud infrastructure. The SARMS provides an adaptive utilisation threshold (CPU and memory) mechanism to dynamically identify the overloaded and underloaded PMs. This system has two steps, migration of VMs from the overloaded PMs to

prevent SLA violations, and consolidation of VMs into a minimum number of active PMs in order to reduce energy consumption. They evaluated the proposed system using the CloudSim based on real workloads from Google and PlanetLab. The obtained results show that the SARMS can achieve performance requirements, while reducing PMs energy consumption and the number of VM migrations. Nevertheless, these approaches do not consider the energy consumption overhead and the costs of VMs consolidation.

Beloglazov and Buyya [127] proposed a technique for dynamic VM consolidation based on CPU utilisation thresholds. This technique focuses on Cloud resource management strategies (e.g., VM migration) with the aim to optimise resource usage and reduce energy consumption, while maintaining the SLAs. It can be achieved by migrating the VMs from the underloaded hosts in order to reduce the number of active hosts and saving energy. To re-allocate the VMs, a Modified Best Fit Decreasing (MBFD) algorithm is used to sort the selected hosts based on their CPU utilisation and energy efficiency. They evaluated the proposed technique through simulations with different types of workloads using PlanetLab servers. The results show that this technique outperforms other migration policies in terms of the number of VM migrations and SLA violation, while showing a similar level of energy consumption. However, the proposed technique lacks to consider the actual cost and power consumption caused by VMs consolidation.

Also, Malekloo et al. [128] introduced a Multi-objective Ant Colony Optimisation (MACO) approach for VMs placement and consolidation algorithms. In this regard, the VMs' placement algorithm aims to minimise energy consumption, CPU resource wastage and communication cost. While, the VM consolidation algorithm aims to reduce SLA violations, VMs migration and the number of active PMs. They evaluated the proposed approach using the CloudSim based on eight performance metrics. The results show that this approach outperforms the other approaches in terms of achieving the balance between energy consumption, system performance and QoS requirements. Yet, this approach focused on minimising PMs energy consumption without taking into consideration the energy consumption incurred by VMs consolidation.

Zhou et al. [16] proposed an adaptive strategy for energy and performance efficient VM consolidation, called (DADTA). The DADTA strategy

aims to minimise energy consumption while satisfying the SLAs in the Cloud data centre. They applied a specific adjustment of thresholds to adapt the dynamic workload changes and then performed VM consolidation by using the DADTA in order to improve the overall optimisation. To evaluate the proposed strategy, a modified prediction model conducted on the CloudSim is used to deal with the time-series data obtained from the Google cluster workload trace, and the findings show that the proposed DADTA outperforms other benchmarks in terms of minimising the PMs energy consumption and SLA violations. In their work, the consolidated VMs are homogeneous and only considers PMs power consumption.

Moreover, Beloglazov and Buyya [115] presented adaptive algorithms for dynamic VM consolidation based on a statistical analysis of historical workload data. Statistical models are used to calculate the upper and lower CPU utilisation thresholds of each host. If the host is determined to be overloaded, one or more VMs are selected to be migrated from the host to another suitable one in order to optimise the resource usage and maintain a high level of SLAs. On the other hand, if the host is determined as underloaded, all hosted VMs are selected to be migrated from the host and switch it to the sleep mode in order to reduce the energy consumption. They evaluated the proposed algorithms through the CloudSim using workload traces from PlanetLab, considering the heterogeneity of PMs and VMs. The results of the experiments show that the proposed algorithms outperform other dynamic VM consolidation algorithms in terms of the level of SLA violations and the number of VM migrations. However, this work only considers PMs energy consumption and does not refer to VMs energy consumption.

The authors in [129], [130] emphasised the importance of taking migration cost into account for a fine-grain VM consolidation strategy. Therefore, Zakarya and Gillam [131] proposed a VM consolidation technique, named a Consolidation with Migration Cost Recovery (CMCR). This technique aims to explore the ability of the VMs to recover their migration costs. In order to achieve that, the VMs should firstly be migrated to an energy efficient host and then continue to run them for a certain period of time. A linear power model is used to identify the power consumption for the target host in order to check the ability of the VMs to recover their migration costs. They evaluated the CMCR through CloudSim using

real workload traces from a Google cluster. The results show that by using the CMCR the majority of the migrated VMs can recover their migration cost. However, their work is applicable only to the hosts that follow a linear power model and does not consider the heterogeneity of PMs or VMs. Similarly, Verma et al. [129] introduced a power-aware application placement framework for virtualised server clusters, called pMapper, which dynamically places the VMs to minimise the power consumption and the migration cost, while meeting the performance requirements. In their framework, they have extended the First Fit Decreasing (FFD) heuristic algorithm in order to migrate the VMs to suitable hosts. This is aimed to minimise the data centre's energy consumption by reducing the number of active hosts, while taking into account the VMs migration cost. They have implemented the pMapper framework on IBM testbed with heterogeneous hosts using a set of benchmark applications. The results show that the pMapper outperforms other power unaware algorithms in terms of minimising the PMs power consumption and VMs migration costs, while meeting the application performance guarantees. However, their framework does not provide any information regarding the migration costs calculation.

2.7.2 Resource Provisioning

Cloud service providers support an on-demand resource provisioning model, called auto-scaling, which provides additional resources requested by applications using vertical and horizontal scaling techniques.

Generally, the auto-scaling can be defined as the ability of a system or users to add and remove resources (such as CPU, memory), which is beneficial for adapting to workload variations and ensuring consistent performance with lower costs [21], [14]. Cloud providers such as Amazon Web Services (AWS) [132] offer this service.

Auto-scaling is a dynamic property for Cloud Computing, and it comes in two types, namely, *vertical* and *horizontal* scaling. The **vertical scaling** is used to add or release virtual resources dynamically (e.g., vCPUs and memory) inside the VMs, whereas, **horizontal scaling** is used to create or delete VMs, all of which were based on application requirements. However, the latter mechanism

may take a few minutes to initiate [17], [18], [133], [102], which may be unsuitable for VMs that need to rapidly scale during the computation [19], [20].

To achieve the scalability of Cloud resources a combination of these two scaling techniques can help to find an optimal scaling strategy [102]. However, most of the vertical and horizontal scaling approaches are *reactive* methods which happen after detecting there are not enough resources for an application [20], [134]. Thus, it is desirable if the methods can be scaled earlier than the time when the workload actually increases. This can be achieved by using *proactive* methods that can predict workloads of applications and scale the resources commensurate with the predicted workload.

A number of solutions have been proposed to support resource elasticity for Cloud applications. For example, Ficco et al. [15] presented a new approach for managing elastic resources reallocation in Cloud infrastructures using the coral-reefs algorithm and game theory optimisation. This approach uses a multi-objective optimisation to maintain customers SLAs, minimise resource consumption and cost during the auto-scaling and migration processes. In their work, the coral-reefs algorithm is used to model the elasticity of Cloud resources, whereas, the game theory is used to optimise the aims of the service provider expressed through resource reallocation strategies with respect to the customer's requirements. The experimental results show that the combination of coral-reefs algorithm and game theory optimisation achieves the elasticity of Cloud resources and leads to significant performance improvements. However, the energy-related cost when performing the auto-scaling and migration is not considered in their approach.

Likewise, Tighe and Bauer [135], [136] developed a rule-based approach that combines the auto-scaling of applications with dynamic VM allocation to match current workload demands and maintain SLA achievement. In their approach, vertical scaling is performed to scale up and down the VMs according to their resource requirements to run applications, as well as, the VMs are consolidated into a minimal number of PMs using live migrations in order to switch off the idle PMs and saving energy costs. As shown on their simulation results, they argued that their combined approach can achieve better application performance with a reduction in VM live migrations compared to the independent approaches. However, their approach only considers the vertical scaling of the

scaled resources and do not consider the prediction of these resources. In addition, the costs of the scaled resources are not considered.

Dawoud et al. [137] proposed a dynamic resource provisioning approach that aims to allocate the minimum resources required to handle the future workload demands while maintaining the Service Level Objectives (SLOs). Their approach includes three controllers for CPU, memory, and application to guarantee efficient resource allocation and optimise the application performance. A linear prediction model is used to predict the future resource requirements for efficient allocation and correspond with the workload demands. They have evaluated the proposed approach using the Xen hypervisor with a synthetic workload, and the results show that their controllers are capable to horizontally scale the VMs to correspond with the workload demands while mitigating the SLO violation. However, their approach only considers the horizontal scaling to cope with VMs workload demands without considering the vertical scaling technique. Also, the energy consumption of provisioned resources is not considered.

Moreover, Meng et al. [138] proposed a joint-VM provisioning approach that estimates the VMs capacity needs through statistical multiplexing principles based on their workload patterns. The main idea of this approach is to borrow unused resources from low utilised VMs and reallocated these resources to the VMs with high utilisation in order to achieve the application performance requirements. The proposed approach is evaluated based on data collected from commercial data centres using simulations. The results demonstrate that the proposed joint-VM provisioning approach has improved the overall resource utilisation by 45% compared to the individual-VM provisioning approaches.

Also, Gandhi et al. [21] investigated the impact of resource auto-scaling on cost, performance and provisioning times for Cloud applications. They employed the Amdahl's Law formula to model service time scaling, the queueing-theoretic concepts to model performance scaling, and a Kalman filtering approach to estimate the performance model parameters. They implemented their approach on OpenStack and the results show the ability of the proposed approach to determining the most cost-effective scaling option for a given workload, considering both horizontal and vertical scaling. However, this

approach does not consider the prediction of resource requirements and their energy consumption when performing the scaling decisions.

Dutta et al. [14] presented an automatic scaling framework called (SmartScale), which uses a combination of horizontal and vertical scaling in order to optimise the resource usage and the reconfiguration cost incurred due to scaling. The SmartScale is a proactive technique that used a polynomial regression in order to estimate the resource requirements to perform the scaling decisions for the next time interval. They evaluated their framework using a real Cloud testbed and the results show that the SmartScale can scale the required resources to run applications with the lowest reconfiguration cost. However, this framework does not consider the power consumption of required resources incurred due to scaling decisions.

2.7.3 Overall Discussion

Cloud resource management has the ability to adapt VMs' consolidation and resource provisioning in order to meet the performance requirements of applications, minimise the operation costs and energy consumptions in Cloud data centres.

Section 2.7 has reviewed the related work on VMs' consolidation and resource provisioning mechanisms in Cloud environments.

In terms of VMs consolidation, a commonly known NP-hard optimisation problem is closely related to it, where the most important objectives are minimising resource usage and energy consumption, while satisfying the SLAs. As discussed in Section 2.7.1, the work in [117], [121], [124] aimed to improve the VMs performance during the migration process, considering the application-aware resource consumption characteristic, but their models only focused on the resource consumption and do not consider the energy consumption overhead of VMs migrations. Moreover, the work presented in [113], [115], [125], [126], [127], [128] mainly focused on dynamic re-allocation of VMs using live migration to increase the overall servers' utilisation and minimise the energy consumption, while maintaining the required QoS. Yet, these approaches focused on minimising PMs energy consumption without taking into consideration the energy consumption incurred by VMs consolidation. Also, the work presented in [129],

[131] have addressed the issue with migration cost, considering the energy consumption at both PMs and VMs levels. Though there are still limited as the model in [129] does not provide any information regarding the migration cost calculation, whereas, the work in [131] is only applicable to the hosts that follow a linear power model and does not consider the heterogeneity of PMs or VMs. Further, the work presented in [118], [123], [16] employed workload prediction models based on historical data to eliminate unnecessary VM migrations, minimise energy consumption and SLA violations. These models focused on improving the performance of Cloud applications by reducing the number of overloaded hosts, but without explicitly considering energy and cost of VMs migrations, as a part of VMs consolidation decision criterion.

In terms of VMs resource provisioning, a fine-grained resource provisioning while ensuring the performance and the SLAs for applications are required, which makes finding the optimal and efficient scaling option a very challenging problem. In Section 2.7.2, the work in [21] investigated the impact of resource auto-scaling on cost, performance, and provisioning times in order to determine the most cost-effective scaling option for Cloud applications. Further, the work presented in [15], [135], [136] combined the auto-scaling of applications with dynamic VM allocation to match current workload demands and maintain SLA achievement. However, the energy consumption related to the auto-scaling and migration decisions is not considered in their approaches. Moreover, the work presented in [137], [138], [14] considered the prediction of resources provisioning to handle the future workload demand while maintaining the SLOs, but these approaches do not consider the power consumption of required resources incurred due to scaling decisions.

Thus, there is still a need for predictive modelling that dynamically supports VMs live migration and auto-scaling decisions, considering the trade-off between cost, power consumption, and performance during service operation, which can help Cloud providers to make better use of their infrastructures and efficiently manage Cloud resources [139], [140].

The following Table 2-4 provides a comparison summary of the closely related work on VMs' consolidation and resource provisioning that considers the workload, energy consumption and cost in Cloud environments, followed by a

comparison summary of the closely related work on the prediction of these mechanisms, as shown in Table 2-5.

Table 2-4: Summary of Existing Models for VMs' Consolidation and Resource Provisioning.

Criteria / by	Workload Consideration		Energy Consumption Consideration		Cost Consideration	
	PMs level	VMs level	PMs level	VMs level	Cost of Migration	Cost of Scaling
Ye et al. [117]	Homogeneous PMs only.	Homogeneous VMs only.	Not considered.	Not considered.	Not considered.	—
Zhao et al. [121]	Heterogeneous PMs.	Heterogeneous VMs.	Not considered.	Not considered.	Not considered.	—
Xu et al. [124]	Homogeneous PMs only.	Heterogeneous VMs.	Homogeneous PMs only.	Not considered.	Not considered.	—
Beloglazov and Buyya [125], Beloglazov et al. [126], Malekloo et al. [128]	Heterogeneous PMs.	Not considered.	Heterogeneous PMs.	Not considered.	Not considered.	—
Farahnakian et al. [113]	Heterogeneous PMs.	Heterogeneous VMs.	Heterogeneous PMs.	Not considered.	Not considered.	—
Beloglazov and Buyya [127], [115]	Heterogeneous PMs.	Not considered.	Heterogeneous PMs.	Not considered.	Considered.	—
Zakarya and Gillam [131]	Homogeneous PMs only.	Homogeneous VMs only.	Homogeneous PMs only.	Homogeneous VMs only.	Considered.	—
Verma et al. [129]	Heterogeneous PMs.	Heterogeneous VMs.	Heterogeneous PMs.	Heterogeneous VMs.	Considered.	—
Ficco et al. [15]	Homogeneous PMs only.	Not considered.	Not considered.	Not considered.	Considered.	Considered.
Tighe and Bauer [135], [136]	Homogeneous PMs only.	Homogeneous VMs only.	Homogeneous PMs only.	Not considered.	Not considered.	—
Gandhi et al. [21]	Homogeneous PMs only.	Homogeneous VMs only.	Not considered.	Not considered.	—	Considered.

Table 2-5: Summary of Prediction Models for VMs' Consolidation and Resource Provisioning.

Criteria / by	Workload Prediction Consideration		Energy Prediction Consideration		Cost Estimation Consideration	
	PMs level	VMs level	PMs level	VMs level	Cost of Migration	Cost of Scaling
Farahnakian et al. [118]	Heterogeneous PMs.	Heterogeneous VMs.	Not considered.	Not considered.	Not considered.	—
Beloglazov and Buyya [123]	Homogeneous PMs only.	Not considered.	Not considered.	Not considered.	Not considered.	—
Zhou et al. [16]	Heterogeneous PMs.	Not considered.	Heterogeneous PMs.	Not considered.	Considered.	—
Dawoud et al. [137]	Homogeneous PMs only.	Homogeneous VMs only.	Not considered.	Not considered.	—	Not Considered.
Meng et al. [138]	Homogeneous PMs only.	Homogeneous VMs only.	Not considered.	Not considered.	Not considered.	—
Dutta et al. [14]	Homogeneous PMs only.	Homogeneous and heterogeneous VMs.	Not considered.	Not considered.	—	Considered (horizontal and vertical scaling).

2.8 Thesis Scope

This thesis aims to enable the awareness of energy consumption, performance variation and cost in a Cloud Infrastructure, as depicted in Figure 2-6. To achieve this aim, an energy-based cost model is firstly developed to attribute the PM's energy consumption to VMs and measures the actual resource usage, power consumption and the total cost for each VM, considering the heterogeneity of the PMs and VMs, as discussed in Chapter 3. An energy-based cost prediction framework is then introduced to predict workload, power consumption and estimate the total cost of the VMs during service operation based on historical workload data, using the ARIMA model and regression analysis, as discussed in Chapter 4. Finally, a proactive performance and energy-based cost prediction framework is introduced to combine VMs consolidation (live migration) and resource provisioning (auto-scaling) in order to design cost-effective strategies, while taking into consideration the trade-off among cost, energy efficiency and performance variation of Cloud services, as discussed in Chapters 5 and 6.

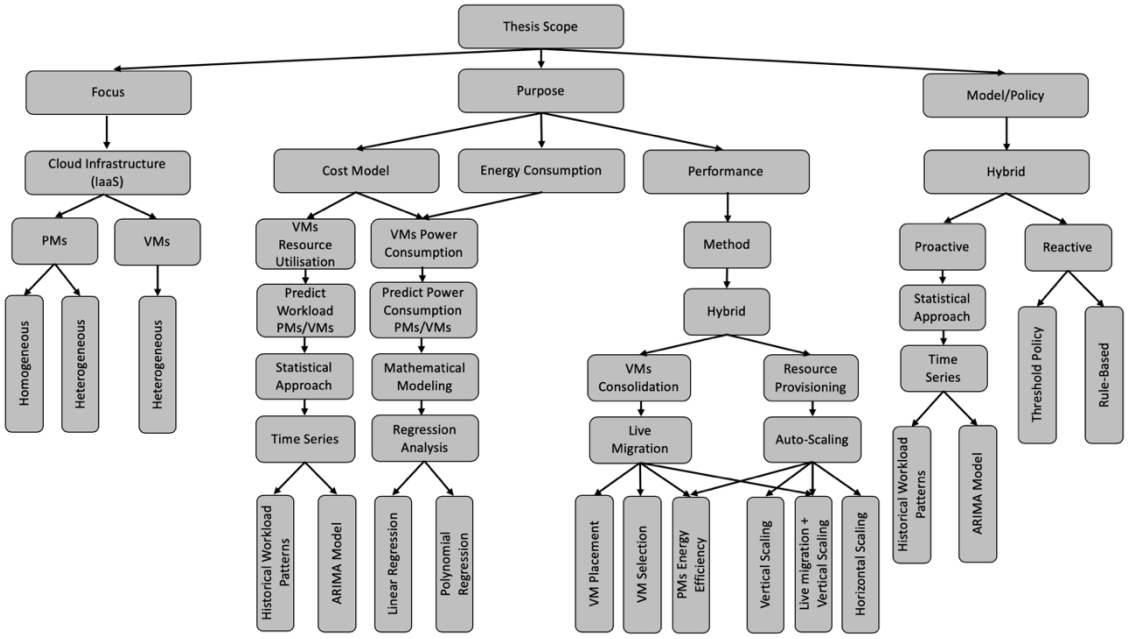


Figure 2-6: Thesis Scope.

2.9 Summary

This chapter has introduced the essential background and the literature related to this research. Firstly, it has discussed some fundamental aspects of Cloud Computing including its definition, system architecture, services types, deployment types and virtualisation technologies. Additionally, it has presented the concepts of Cloud applications and their workload patterns as well as related benchmarks, followed by a description of the pricing models in Cloud Computing. Secondly, it has reviewed the literature on the energy-related cost issues in Cloud Computing, as well as a comparison summary of the closely related work of this research has been presented. Thirdly, it has highlighted the prediction models related to predicting the workload, energy consumption and cost of Cloud services, in addition to a summary discussion of the closely related work has been introduced. This chapter has finally concluded with a discussion of the existing work on Cloud resource management, including VMs consolidation and resource provisioning, along with a comparison summary of the closely related work of this research and a presentation of the thesis scope.

Chapter 3. System Architecture and Energy Cost Modelling

3.1 Overview

In this chapter, definitions and assumptions considered in this thesis are given in Section 3.2. Section 3.3 presents the system architecture that supports energy, performance and cost awareness of Cloud infrastructure services, followed by the descriptions of the required components and their interactions within the proposed architecture. Section 3.4 presents an energy-based cost model that considers energy consumption as a key parameter with respect to the actual resource usage and the total cost of the VMs. This chapter concludes by discussing early experiments conducted on a Cloud testbed to validate the ability of a proposed model of estimating the actual total cost of the VMs based on their actual resource usage with consideration of their energy consumption, as presented in Sections 3.5 and 3.6.

3.2 Definitions and Assumptions

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

- This research makes abstraction of the type of Cloud applications. Yet, the modelling and prediction in this research are driven through Cloud application workload patterns, in the sense that it considers Cloud applications having repeated historical workload patterns, **periodic** only, when modelling and predicting the VMs workload and energy consumption.
- This research considers heterogeneous VMs. The term **homogeneous** VMs refers to the VMs having the same size in terms of the number of vCPUs and RAM, while the term **heterogeneous** VMs refers to the VMs having different sizes based on their number of vCPUs and RAM.
- **VM workload** is represented as (CPU, RAM, disk and network), when modelling and predicting the VM workload and cost.

- **Virtual CPUs (vCPUs)** utilisation represents the workload of the VM, only when modelling and predicting VM energy consumption. It is measured in percentage unit (%).
- **PM CPU utilisation** represents the workload of the PM, when modelling and predicting PM energy consumption. It is measured in percentage unit (%).
- **PM power consumption** represents the actual or predicted power consumption of the PM at a given point in time, when modelling or predicting VM power consumption, which includes the idle and active power. It is measured by Watt (W).
- The **idle energy** of the PM is attributed to homogeneous and heterogeneous VMs by considering the size of each VM in terms of the vCPUs assigned to it.
- The **active energy** of the PM is attributed to homogeneous and heterogeneous VMs by considering the VM CPU utilisation and number of vCPUs assigned to each VM.
- **VM power consumption** represents the attributed power consumption of the VM at a given point in time, when modelling or predicting, which includes (the idle and active power). It is measured by Watt (W).
- **Power** is the rate of electrical usage when performing a work at an instant of time and it measured by Watt (W). **Energy** is the averaged power consumption over a period of time to deliver a work and is measured by Kilowatt-Hour (kWh).
- **VM total cost** represents the cost of VM workload (including CPU, RAM, disk and network) along with the cost of VM energy consumption (driven only through the CPU utilisation) for a period of time, when modelling and estimating the total cost of the VM. It is charged in British Pound Sterling (GBP/£). However, there are various costs incurred by Cloud providers such as software licenses, IT support, cooling and maintenance, which are out of the scope of this research. In addition, other system resources such as memory, disk and network, as well as the hypervisor and context switches consume energy, but this research considers the energy-related to CPU utilisation only, see Section 3.4.

- When the VMs are idle and no tasks have been assigned to them, they have to share the idle power of the host based on their size (the number of vCPUs assigned to each VM); the **cost of idle energy** is considered in the calculation of the VM total cost.
- The research presented in this thesis makes use of **a local Cloud testbed** with a limited scale (4 PMs and 3 VMs are only used, see Section 3.5).

3.3 Proposed System Architecture

The proposed Cloud system architecture is based on the three standard layers (discussed in Chapter 2), which are Software as a Service (SaaS) where the service creation takes place, Platform as a Service (PaaS) where the service deployment takes place, and Infrastructure as a Service (IaaS) where the service operation takes place, as shown in Figure 3-1.

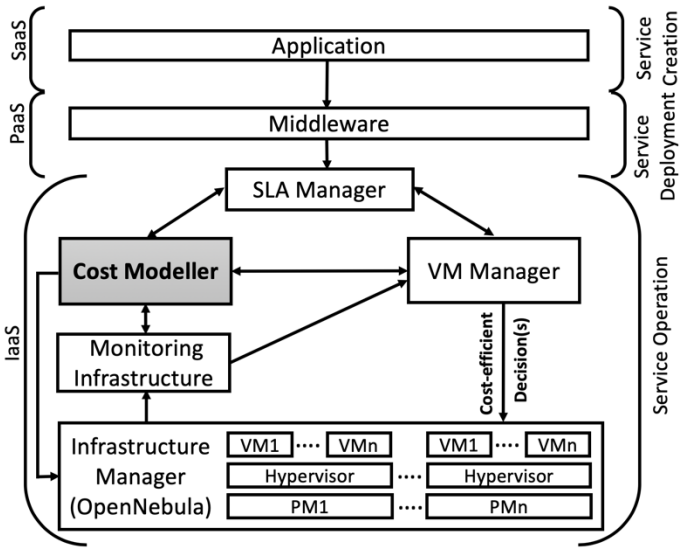


Figure 3-1: System Architecture.

This proposed architecture summaries the high-level details of these three layers and mainly focuses on the IaaS layer where the service operation takes place. In the IaaS layer, the admission, allocation and management of VMs are performed through the interaction between a number of components. These components and their interactions within this architecture, are discussed in Section 3.3.1. The highlighted component *Cost Modeller* is the main component of interest including the other contributions of this thesis. The overall aim of the

Cost Modeller is to advance beyond the state of the art by considering the awareness of energy consumption, performance variation and total cost of Cloud infrastructure services.

3.3.1 Key Components and Interactions

As depicted in Figure 3-1, the IaaS layer in the proposed architecture consists of a number of components, mainly the Service Level Agreement (SLA) Manager, Virtual Machine Manager (VMM), Infrastructure Manager (IM), Monitoring Infrastructure (MI) and *Cost Modeller*. The *Cost Modeller* is the main component within this architecture, it interacts with other components with the aim to support the effectiveness of the proposed architecture. In order to achieve that, the SLA manager continually monitors SLA conformance and interacts with the VMM to determine the SLA offers. The role of the VMM is to react to any periodical events such as VMs consolidation and resource provisioning in order to optimise resource management during service operation. Thus, the VMM would need to utilise the *Cost Modeller* as well as the essential data from MI in order to handle these events efficiently. In this regard, the *Cost Modeller* would help to provide cost-efficient decisions related to the VMs based on their resource usage and power consumption, then send to the VMM to perform. Further details of these components and the role of *Cost Modeller* plays in each, are discussed next.

3.3.1.1 SLA Manager

The SLA Manager is responsible for monitoring and measuring the application SLA's agreed terms at the IaaS layer. This component interacts with the *VMM* to check the availability and capability of resources to determine the SLA offers as well as interacts with the *Cost Modeller* to assign the cost of the offered terms.

3.3.1.2 VM Manager

The VMM component is responsible for managing the VMs at service operation level. This component considers an efficient resource management decision(s) such as VMs consolidation and resource provisioning in order to improve resource usage and reduce the energy cost, and consequently the total cost of

Cloud services. In the case of service performance degradation, this component will interact with the *Cost Modeller* to request measures or predictions related to the resource usage, power consumption and cost that VMs would incur on any particular host. In this research, the service performance refers to the resources required to run applications in an efficient way, in terms of resource availability, energy efficiency and cost.

3.3.1.3 Infrastructure Manager

The IM manages the entire physical infrastructure that includes e.g., processors, memory, storage devices, networking and hardware energy meters. In this component, the VMs are managed by the Hypervisor, which allows sharing of the physical resources among the VMs.

3.3.1.4 Monitoring Infrastructure

The main role of this component is to monitor the PMs and VMs resource usage (e.g., CPU, memory, network and disk), PMs' energy consumption (e.g., Watts-hour) and performance-related metrics (e.g., CPU utilisation and memory usage) during the execution of the applications at the service operation level.

3.3.1.5 Cost Modeller

The overall aim of this component is to: 1) enable the awareness of energy consumption, performance variation and total cost of the VMs at the operational level, and 2) predict the workload and power consumption as well as estimate the total cost of the VMs incurred by different resource management decisions (e.g., VMs re-allocating, live migration and auto-scaling). Therefore, this component supports:

- **Energy-based Cost Model** that provides measuring the actual resource usage, power consumption and total cost relating to the VMs. The details of this model will be discussed in Section 3.4.
- **Energy-based Cost Prediction Framework** that predicts the resource usage, power consumption and estimates the total cost for the VMs. The details of this framework will be discussed in Chapter 4.

- **Performance and Energy-based Cost Prediction Framework** that supports actuators (e.g., VMs re-allocating, live migration and auto-scaling) to tackle the performance variation and attempt to get the performance to the expected level with minimal impact on cost. Furthermore, the proposed framework (in Chapter 4) is used in this context to predict the PMs/VMs workload and power consumption as well as estimate the total cost of the VMs incurred by live migration and auto-scaling decisions. The details of this framework will be discussed in Chapter 5.
- **A Hybrid Approach for Performance and Energy-based Cost Prediction** that dynamically supports decision-making regarding auto-scaling and live migration costs, while at the same time being aware of the impact on other quality characteristics such as energy consumption and performance of the application. In this hybrid approach, the proposed framework (in Chapter 5) has been extended by integrating auto-scaling with live migration in order to perform the most cost-effective decision to handle the service performance variation. The details of this approach will be discussed in Chapter 6.

3.4 Energy-based Cost Model

Modelling a new cost mechanism for Cloud services that can be adjusted to the actual energy costs has attracted the attention of many researchers. With the increasing cost of electricity [11], Cloud providers consider energy consumption as one of the biggest cost factors to be maintained within their infrastructures [1]–[3], [83].

In a Cloud environment, each PM can run a single VM or multiple VMs simultaneously. These VMs can be homogeneous or heterogeneous based on their characteristics, for example, the number of Virtual CPUs (vCPUs) and memory size. Thus, these parameters should be taken into consideration along with their power consumption when modelling and identifying the total cost for the VMs.

Most Cloud infrastructure providers charge their customers for the offered services on a time-based fee [82] regardless of the actual resource usage [23]

and consideration of energy consumption [23], [10]. Therefore, an energy-based cost model that considers energy consumption as a key parameter with respect to the actual resource usage and the total cost is proposed in this thesis. This model accounts based on the actual resource usage (e.g., vCPUs, memory, network and disk) taking into account the power consumption of the VMs.

The PMs power consumption can be directly measured through monitoring tools either internal such as Running Average Power Limit (RAPL) [141] and Intelligent Platform Management Interface (IPMI) [142] or external such as Watt's Up Power Meter [143]. Unlike PMs, a VMs' power consumption is difficult to identify and cannot be directly measured as they do not have physical interfaces to plug in any of the power meters for example. Instead, the power consumption of VMs can be gathered from their underlying PMs, which is still difficult to achieve [144], [145].

Many of the existing approaches model and identify the energy consumption in PMs, as presented in [2], [146], [147] and the energy consumption in VMs, as proposed in [131], [148], by considering only the CPU utilisation. Therefore, understanding how resource usage affects the power consumption is required. An experimental study that investigates the effect of the resource usage (e.g., CPU, memory, disk and network) on the power consumption is presented in Section 3.6. The findings show that the CPU utilisation is highly correlated with the power consumption, as supported in other work, for example [2], [146], [149], [108], [118], [113]. Thus, the proposed model in this thesis follows the same approach and takes into account the CPU utilisation only when modelling and identifying the energy consumption for the VMs [144].

The energy-based cost model introduced in this chapter works by firstly measuring the VMs workload as well as the PMs energy consumption through a monitoring system [150]. After that, this model would attribute the PM's energy to the VMs in order to estimate the energy consumption for each VM. Then, the VMs total cost can be obtained based on the measured workload and energy consumption for each VM. In order to achieve that several steps are required:

Step 1: The VMs workload (the actual resource usage including vCPUs, memory, network and disk) is measured through a monitoring system [150] for

each VM. Similarly, the PMs power consumption can be directly measured through a monitoring system [150] for each PM, as long as each of the PM has a Watts Up [143] meter attached to it.

Step 2: After the VMs workload and PMs power consumption are measured, the second step is to attribute the PM power consumption to the new requested VM and to the VMs already running on the PM. Hence, the power consumption for the new VM can be done in two parts: 1) VMs idle power consumption, $\mathrm{VM}x_{IdlePwr}$ based on the number of vCPUs assigned to each VM [144], as shown in Equation (3.1). The idle energy of the PM is attributed to homogeneous and heterogeneous VMs by considering the size of each VM in terms of the vCPUs assigned to them, and 2) VMs active power consumption, $\mathrm{VM}x_{ActivePwr}$ based on the VM CPU utilisation as well as the number of vCPUs assigned to each VM [144], as shown in Equation (3.2). The active energy of the PM is attributed to heterogeneous and homogeneous VMs by considering the VM CPU utilisation and number of vCPUs assigned for each VM.

$$\mathrm{VM}x_{IdlePwr} = PMx_{IdlePwr} \times \left(\frac{\mathrm{VM}x_{ReqvCPUs}}{\sum_{y=1}^{VMcount} \mathrm{VM}y_{ReqvCPUs}} \right) \tag{3.1}$$

where $PMx_{IdlePwr}$ is the idle power consumption of the PM where the VMs are hosted; $\mathrm{VM}x_{ReqvCPUs}$ is the number of the vCPUs assigned to the given VM; VM_{Count} is the number of VMs running on the same PM; and $\sum_{y=1}^{VMcount} \mathrm{VM}y_{ReqvCPUs}$ is the number of vCPUs assigned to a number of the VMs set hosted by the same PM.

$$\mathrm{VM}x_{ActivePwr} = (PMx_{Pwr} - PMx_{IdlePwr}) \times \left(\frac{\mathrm{VM}x_{(Util \times ReqvCPUs)}}{\sum_{y=1}^{VMcount} \mathrm{VM}y_{(Util \times ReqvCPUs)}} \right) \tag{3.2}$$

where PMx_{Pwr} is the total power consumption of the PM, from which the PM's idle power $PMx_{IdlePwr}$ is deducted to identify the PM's active power; $\mathrm{VM}x_{(Util \times ReqvCPUs)}$ is the VM CPU utilisation times the number of vCPUs assigned to the given VM; and $\sum_{y=1}^{VMcount} \mathrm{VM}y_{(Util \times ReqvCPUs)}$ is the VMs CPU utilisation times the number of vCPUs for a set of VMs hosted by the same PM.

Thus, the total power consumption, $\mathrm{VM}x_{Pwr}$, for each VM at any given time can be identified by summing up both idle and active power consumption [144], as shown in Equation (3.3) and Equation (3.4), respectively.

$$\mathrm{VM}x_{Pwr} = PMx_{IdlePwr} \times \left(\frac{\mathrm{VM}x_{ReqvCPUs}}{\sum_{y=1}^{VMcount} \mathrm{VM}y_{ReqvCPUs}} \right) + (PMx_{Pwr} - PMx_{IdlePwr}) \times \left(\frac{\mathrm{VM}x_{(Util \times ReqvCPUs)}}{\sum_{y=1}^{VMcount} \mathrm{VM}y_{(Util \times ReqvCPUs)}} \right) \quad (3.3)$$

which is equal to:

$$\mathrm{VM}x_{Pwr} = \mathrm{VM}x_{IdlePwr} + \mathrm{VM}x_{ActivePwr} \quad (3.4)$$

where $\mathrm{VM}x_{Pwr}$ is the total power consumption for one VM (idle and active power) measured by Watt. The $PMx_{IdlePwr}$ is the idle power consumption and PMx_{Pwr} is the total power consumption for a single PM. $\mathrm{VM}x_{ReqvCPUs}$ is the requested number of vCPUs and $\sum_{y=1}^{VMcount} \mathrm{VM}y_{ReqvCPUs}$ is the total number of vCPUs for all VMs on the same PM. $\mathrm{VM}x_{(Util \times ReqvCPUs)}$ is the VM CPU utilisation times the number of vCPUs assigned to the given VM; and $\sum_{y=1}^{VMcount} \mathrm{VM}y_{(Util \times ReqvCPUs)}$ is the VMs CPU utilisation times the number of vCPUs for a set of VMs hosted by the same PM.

Hence, the energy-based cost model can fairly attribute the idle and active energy consumption of a PM to VMs with the same or different sizes in terms of the allocated vCPUs for each VM. As the VMs are heterogeneous in terms of size, they consequently have different attribution of the idle and active energy consumption, which fairly corresponds to their size. For instance, when both a small VM with 1 vCPU and a large VM with 4 vCPUs are being fully utilised on the same PM, the large VM would be attributed about four times the amount of energy consumption as compared to the small VM (see Section 3.6.2). Thus, this model can help to assess how the power consumption of the PMs is attributed to the VMs based on the actual physical CPU utilisation used by each VM. Also, this model can explore the impact of the actual resource usage on the power consumption of the VMs, especially when the VMs are running on different hosts with different energy characterisation, as presented in Section 3.6.2.

After identifying the power consumption for each VM, the conversion of power to energy is required using Equation (3.5), since the energy providers charge by Kilowatts per hour (kWh).

$$\mathrm{VM}x_{Energy} = \frac{\mathrm{VM}x_{Pwr}}{1000} \times \frac{Time_s}{3600} \quad (3.5)$$

where $\mathrm{VM}x_{Energy}$ is the energy consumption of the VM, measured by kWh. $\mathrm{VM}x_{Pwr}$ is the total power consumption for one VM (idle and active power) measured by Watt (W) times the period of time $Time_s$, measured by second.

Step 3: The final step in this model is to estimate the total cost of the VM based on the actual resource usage from *Step 1* and power consumption from *Step 2*. The following Equation (3.6) is used:

$$\begin{aligned}\mathrm{VM}x_{TotalCost} = &\left(\left(\mathrm{VM}x_{ReqvCPUs} \times \frac{VMx_{Util}}{100}\right) \times (Cost\ per\ vCPU\ \times\ Time_s)\right) \\ &+ \left(\mathrm{VM}x_{RAMUsage} \times\ (Cost\ per\ GB\ \times\ Time_s)\right) \\ &+ \left(\mathrm{VM}x_{DiskUsage} \times\ (Cost\ per\ GB\ \times\ Time_s)\right) \\ &+ \left(\mathrm{VM}x_{NetUsage} \times\ (Cost\ per\ GB\ \times\ Time_s)\right) \\ &+ \left(\mathrm{VM}x_{Energy} \times\ Cost\ per\ kWh\right)\end{aligned} \tag{3.6}$$

where $\mathrm{VM}x_{TotalCost}$ is the total cost of a single VM. The $\mathrm{VM}x_{ReqvCPUs}$ is the number of requested vCPUs and $\mathrm{VM}x_{Util}$ is the CPU utilisation for each VM times the cost for requested vCPUs for a period of time. $\mathrm{VM}x_{RAMUsage}$ is the resource usage of RAM times the cost for that resource for a period of time. We consider the similar notation for the disk and network resources. $\mathrm{VM}x_{Energy}$ is the energy consumption of the VM times the energy cost as considered by the energy providers.

3.5 Early Implementation

In order to obtain an early evaluation of the proposed energy-based cost model, a number of experiments have been conducted on an existing Cloud testbed, available at the University of Leeds. The details of this testbed and how it monitors the resource usage and energy consumption at the PM and VM levels will be discussed next.

3.5.1 Cloud Testbed

The Cloud testbed consists of a cluster of eight commodity Dell servers, and each one of these servers has Linux CentOS version 6.6 installed as its operating

system (OS). Four of these servers with four core X3430 and eight core E31230 V2 Intel Xeon CPU were used. Also, each server has a total of 16GB RAM and 250GB up to 500GB 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 3-2. The testbed utilises a Virtual Infrastructure Manager (VIM), OpenNebula [49] version 4.10.2, Privileged Virtual Machine (PVM) to manage and monitor the Virtual Machine Manager (VMM), the testbed uses Kernel-based Virtual Machine (KVM) [38] hypervisor version 4.0.1 along with the Linux Kernel version 2.6.32.24.

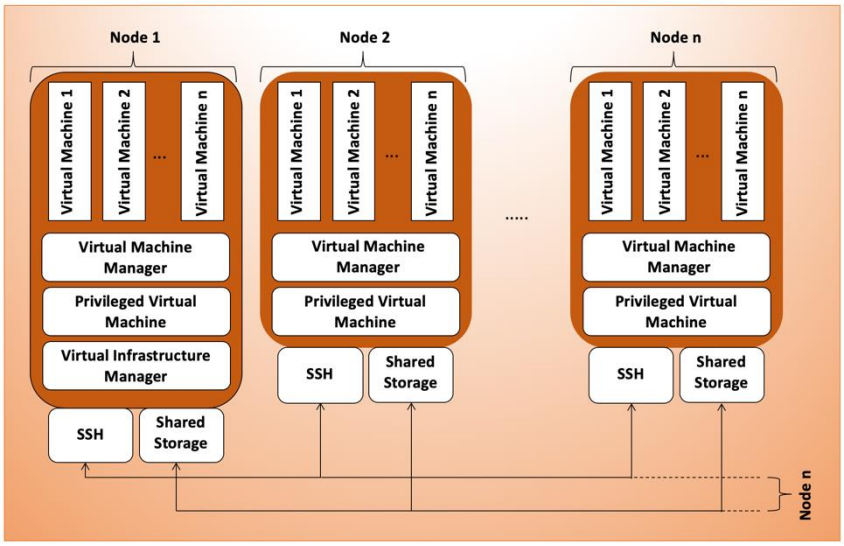


Figure 3-2: Cloud Testbed Architecture.

3.5.2 Monitoring Infrastructure

The resource usage and energy monitoring on the Cloud testbed is depicted in Figure 3-3. At the physical host level, each of the PM has a Watts Up meter [143] attached to directly measure the power consumption on a per second basis for each PM. The measured power values are then pushed to Zabbix [150], which is the monitoring infrastructure tool used on this testbed. Additionally, Zabbix also monitors the resources usage such as CPU, memory, network and disk, for each of the running PMs and VMs. The PMs power usage along with the VMs resource usage are sent to the *Cost Modeller*, which is responsible for measuring energy consumption along with the total cost for the VMs, as described in Section 3.6.

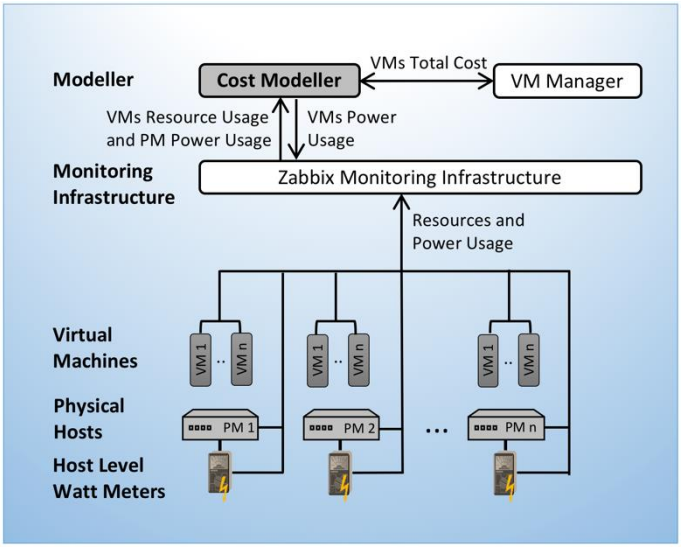


Figure 3-3: Monitoring Infrastructure.

3.5.3 Specifications of PMs and VMs

As explained earlier, the testbed has a cluster of commodity Dell servers, and the following Table 3-1 summarises the configurations of the four PMs considered in this thesis. Hosts A and B are considered in the experiments conducted in this Chapter and Chapter 4. Hosts A, B, C, and D are considered in the experiments conducted in Chapters 5 and 6, respectively.

Table 3-1: Configurations of the PMs.

Hostname	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	An 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)	500GB (Model Number: ST1000NM0033)
Host C	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 D	A four core X3430 Intel Xeon CPU (default clock speed of 2.40GHz)	Total of 16GB of RAM (four modules of 4GB DDR3 at 1333MHz)	500GB (Model Number: WD5003ABYX)

In terms of the VMs considered in the experiments presented in this thesis, Table 3-2 summarises the configurations of the VMs. Rackspace [151] is used as a reference for the VMs configurations, as it provides a wide range of VM types, which gives the customers lots of flexibility to meet their needs. Three types of VMs, small, medium and large are used in this thesis with different

capacities. The cost of the virtual resources is set according to ElasticHosts [152] and VMware [153], whereas they describe a service cost breakdown in detail as follows: 1 vCPU = £0.008/hr, 1 GB Memory = £0.016/hr, 1 GB Storage = £0.0001/hr, 1 GB Network = £0.0001/hr; and the cost of energy = £0.14/kWh [154].

Table 3-2: Configurations of the VMs.

Instance Type	vCPU	Memory	Disk	Network
Small VM	1 vCPU	1GB	20GB	1GB
Medium VM	2 vCPUs	2GB	20GB	1GB
Large VM	4 vCPUs	4GB	20GB	1GB

3.6 Experiments and Evaluation

3.6.1 Design of Experiments

A number of direct experiments have been conducted on the Cloud testbed. The overall aim of these experiments is to evaluate the capability of the energy-based cost model for measuring the actual resource usage, power consumption and total cost at the VM level. Furthermore, the proposed model focuses on overall cost savings of the VMs that can be obtained when running the VMs on different hosts have different energy characterisation.

In order to design such experiments, a software tool called *Stress-ng* [73] is used to induce the workload on the VMs/PMs in different selectable ways. The aim is to generate synthetic periodic workload patterns to represent real workload patterns of Cloud applications by stressing all the resources, e.g., CPU, RAM, disk and network on different types of VMs to their full utilisation. Also, *Stress-ng* is used in order to investigate the relation between CPU utilisation and power consumption. All the experiments are repeated five times 30 minutes each and the statistical analysis is performed to consider the mean values of the results and eliminate any anomalies due to the dynamicity of the cloud. Note that each experiment is set to run for 30 minutes in order to ensure it runs long enough and produce relevant data to support the workload patterns.

The following experiments have been designed to show various aspects of energy consumption at the PM and VM levels. This way can help to assess

how the power consumption of the PMs is attributed to the VMs and explore the impact of the actual resource usage and power consumption on the VMs total cost when being run on different hosts.

3.6.2 Evaluation

The conducted experiments show the results for three types of VMs, small, medium and large when being run on different PMs, (Host A and Host B), having different characteristics in terms of resources and energy consumption.

The aim of these experiments is to evaluate the capability of the proposed energy-based cost model to measure the actual resource usage, power consumption and total cost for a number of VMs when being run on heterogeneous PMs.

In terms of measuring the VMs resource usage, Zabbix [150] monitoring tool system is used, Figures 3-4, 3-6 and 3-8 depict the results of the actual VMs workload, including CPU, RAM, disk and network usage for the VMs. As mentioned earlier, all the VMs workload are repeated five times 30 minutes to perform the statistical analysis by considering the mean values of the results and eliminate any anomalies. The vertical error bars illustrate the standard deviation from the mean values. Based on the measured workload for each VM, their power consumption is also measured via the remaining steps within the proposed model. Figures 3-5, 3-7 and 3-9 show the actual results of the power consumption for all VMs (small, medium and large), respectively, when being run on different PMs (Host A and Host B). As a result, the power consumption attribution for each VM is affected by the variation in the CPU utilisation of all VMs.

The conducted experiments have shown the energy consumption attribution for three heterogeneous VMs running on Host A and Host B, and revealed that they can have different attribution of energy consumption based on the power characteristics of the underlying 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 3-5, 3-7 and 3-9, respectively. Hence, enabling energy-awareness at the VM level can help Cloud service providers

monitor the energy consumption of the VMs and, if necessary migrate the VMs to another host to maintain their energy goals as an example.

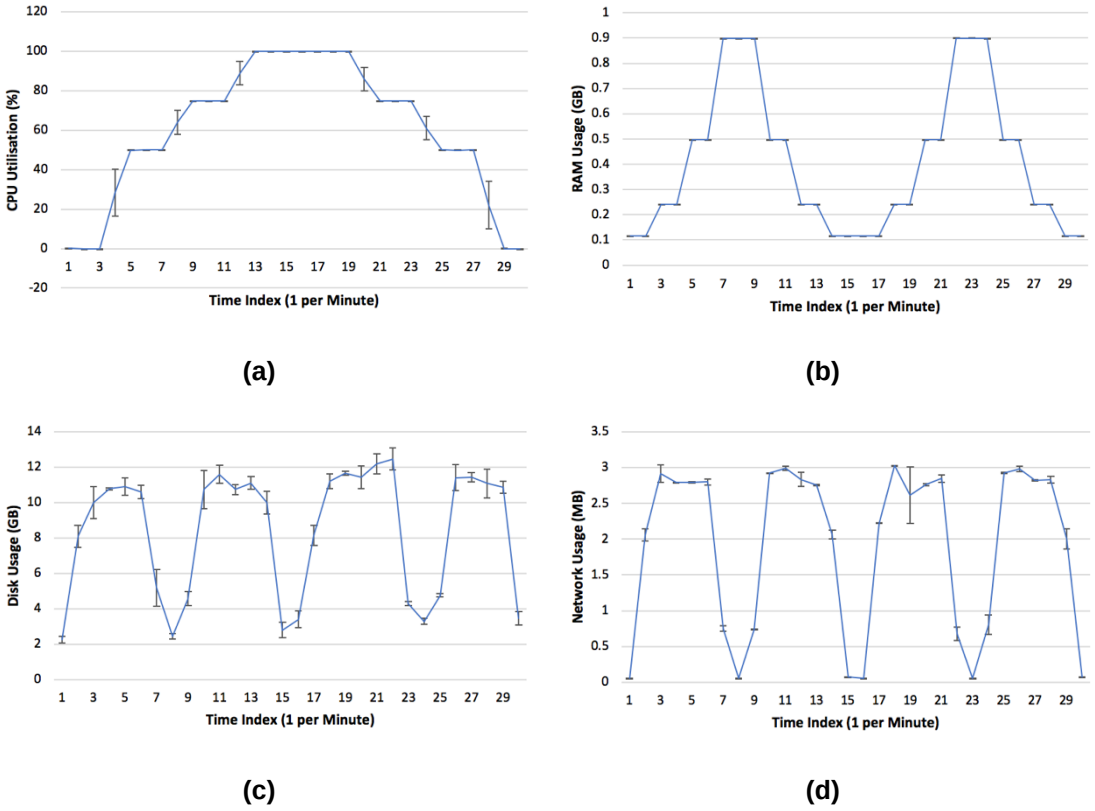


Figure 3-4: The Workload Results for Small VM (for 30 minutes).

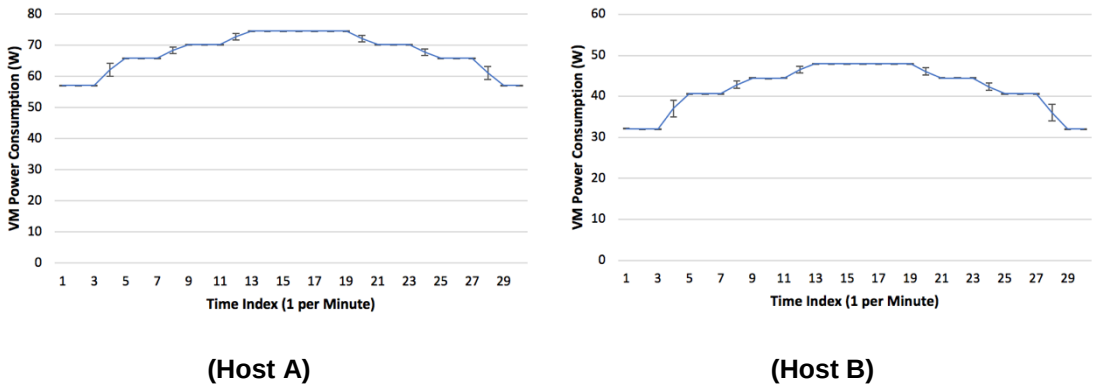
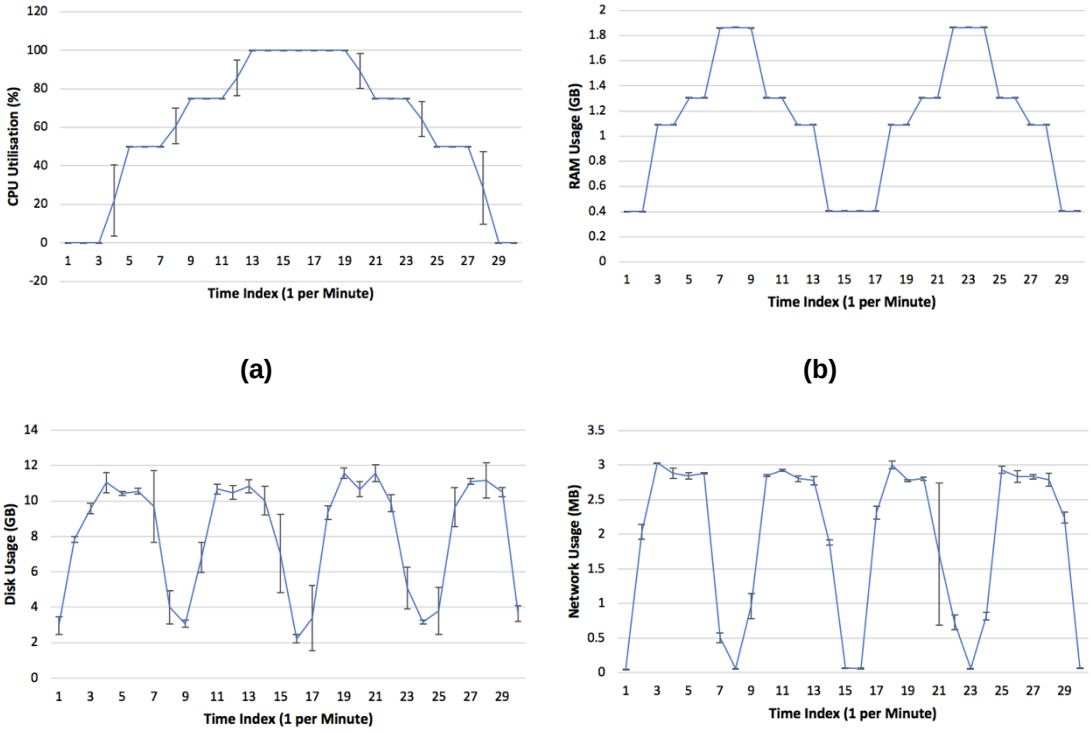


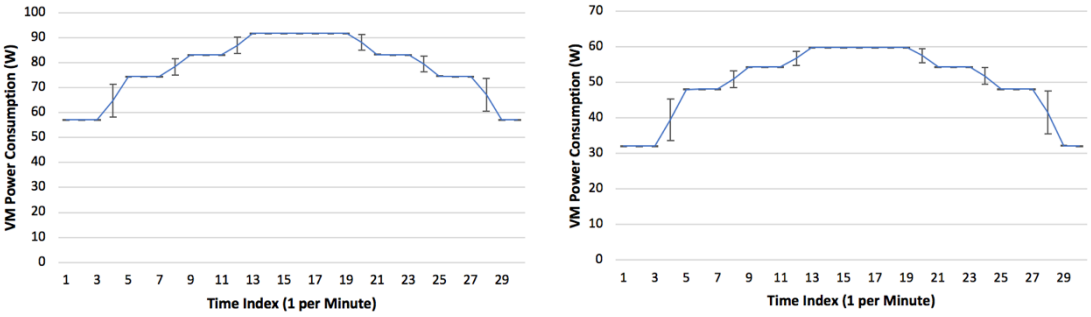
Figure 3-5: Power Consumption Small VM on Host A and Host B.



(a) **(b)**

(c) **(d)**

Figure 3-6: The Workload Results for Medium VM (for 30 minutes).



(Host A) **(Host B)**

Figure 3-7: Power Consumption Medium VM on Host A and Host B.

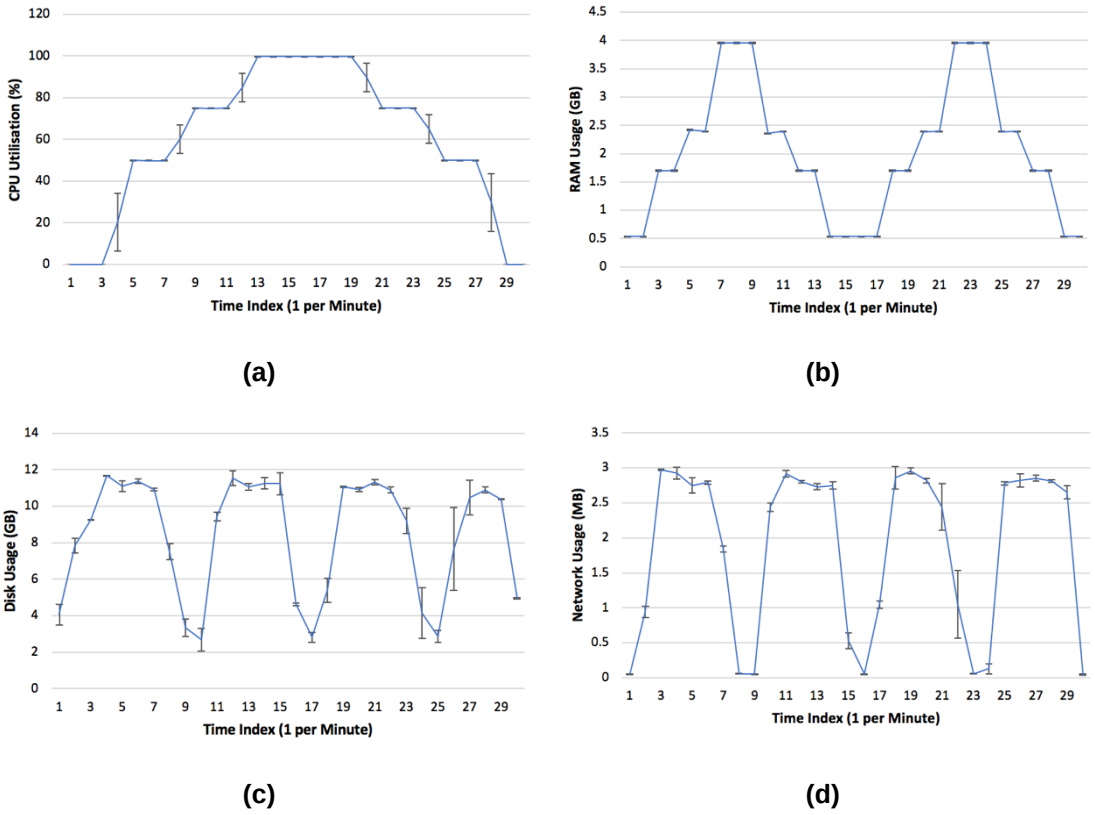


Figure 3-8: The Workload Results for Large VM (for 30 minutes).

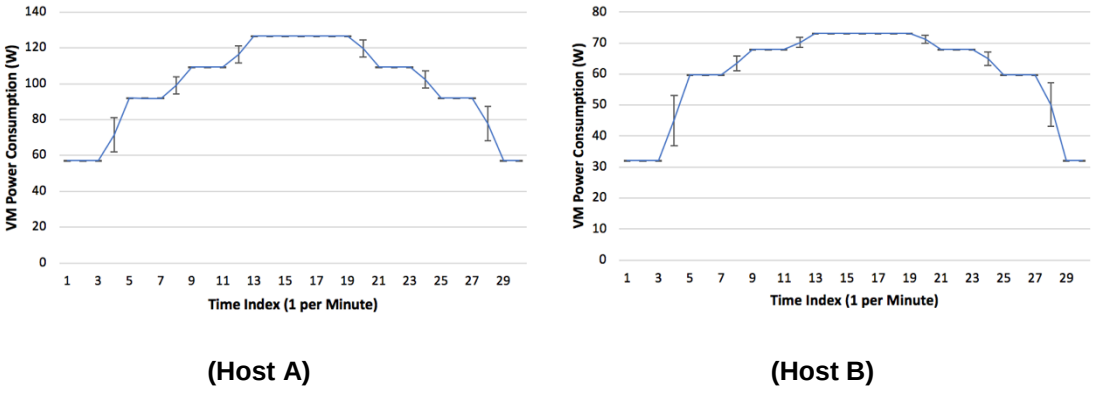


Figure 3-9: Power Consumption Large VM on Host A and Host B.

For clarifying how the proposed model can fairly attribute the PMs power consumption to the VMs, Figures 3-10 and 3-11 show the distribution of the PMs mean power consumption to all three VMs over time (30 minutes) when being run on Host A and Host B, respectively. As designed, all the VMs are idling for the first 15 minutes and actively running with 80% of CPU utilisation for the remaining 15 minutes [144].

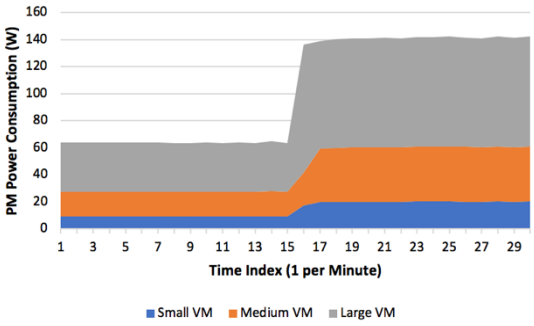


Figure 3-10: PM Mean Power Consumption Attributed to each VM - Host A.

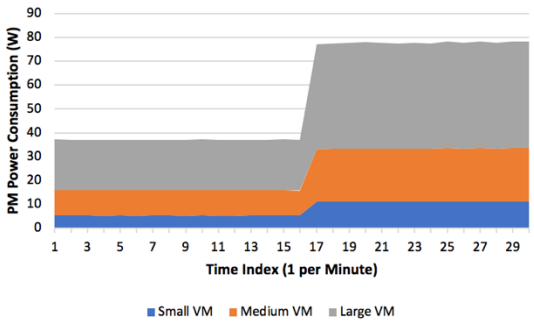


Figure 3-11: PM Mean Power Consumption Attributed to each VM - Host B.

Figures 3-12 and 3-13 show the mean energy consumption per VM in terms of their idle, active and total energy. As the VMs are heterogeneous in terms of size, they consequently have different attribution of the idle and active energy consumption, which fairly corresponds to their size. The energy consumption of a small VM is about twice smaller than a medium VM and about four times smaller than the large VM, which is fairly based on their CPU utilisation and sizes defined by the number of vCPUs each VM has. Further, the conducted experiments have revealed that a considerably large portion of the VMs total energy resides on their idle energy, which is being attributed from the idle energy of the underlying PM. Thus, attributing the PMs 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.

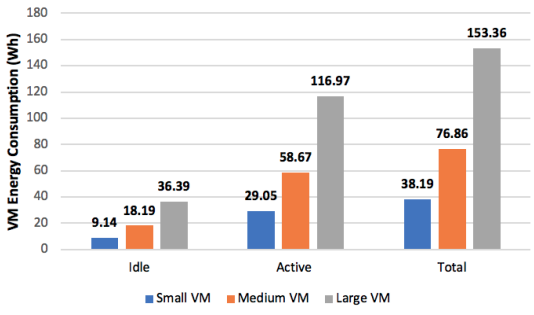


Figure 3-12: Mean Energy Consumption per VM (for 30 minutes) - Host A.

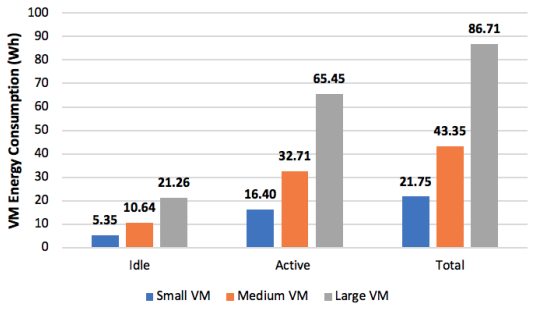


Figure 3-13: Mean Energy Consumption per VM (for 30 minutes) - Host B.

The proposed model is also capable of obtaining the total cost for a number of VMs hosted/running on different PMs as shown in Figure 3-14, which presents the total cost for all the VMs running on different PMs (Host A and Host

B). As the VMs are heterogeneous, the costs of VMs are consequently different. The cost of a small VM is about twice smaller than a medium VM and four times smaller than a large VM when there are running on both Host A and Host B, which is fairly based on their actual resource usage and energy consumption by each VM. The energy efficiency of Host B plays an important role to reduce the total cost (Cost Saving) of the VMs as compared to Host A, as shown in Figure 3-15.

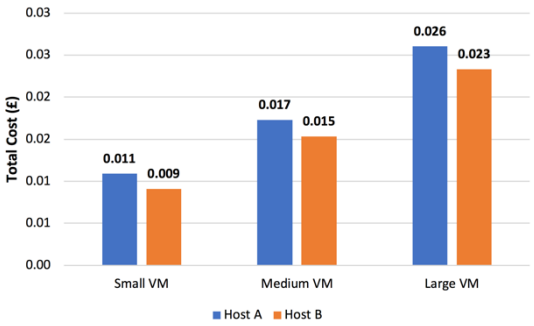


Figure 3-14: The VMs Total Cost on Host A and Host B.

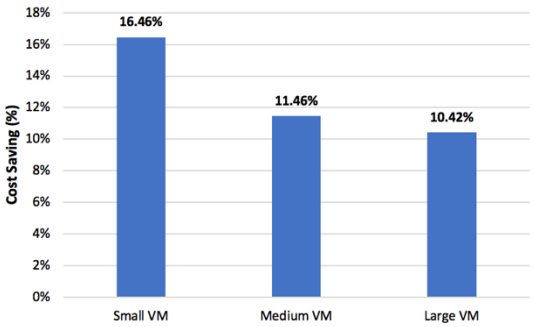


Figure 3-15: The VMs Cost Saving on Host B.

Despite the combination of different types of VMs running on different PMs, the results indicate that the proposed model is capable of estimating the actual total cost for a number of VMs based on their actual resource usage with consideration of their energy consumption.

As mentioned earlier in Section 2.4, the work presented in this thesis is primarily focused on the costs of the cloud infrastructure that are associated with resources along with their energy consumption. However, there are various costs incurred by Cloud providers such as software licenses, IT support, cooling and maintenance, which are out of the scope of this research.

3.7 Summary

To enable performance, cost and energy awareness in a Cloud environment, a system architecture along with the main component *Cost Modeller* are proposed in this thesis. Furthermore, an energy-based cost model that considered energy consumption as a key parameter with respect to the actual resource usage and the total cost of heterogeneous VMs during service operation has been

presented and discussed comprehensively in this chapter. A number of direct experiments were conducted on a Cloud testbed to evaluate the ability of the proposed model to fairly attribute the PM’s energy consumption to VMs and estimates the actual cost for different VMs based on their resource usage with consideration of their energy consumption.

Additionally, extra care has been put into the overall process from experimental design, implementation, data collection and data analysis to ensure it is thorough and consistent. Hence, a statistical analysis has been performed to consider the mean values and the standard deviation of the results in order to eliminate any anomalies. This has helped the proposed model validation and gave confidence in its output.

Chapter 4. Energy-based Cost Prediction Framework

4.1 Overview

In this chapter, an energy-based cost prediction framework that aims to estimate the total cost of VMs by considering their resource usage and power consumption is presented in Section 4.2. This framework works by predicting the VMs' workload based on historical workload patterns and correlating the predicted VMs workload with physical resources in order to estimate the power consumption of the VMs. It then estimates the VMs' total cost accordingly. A number of experiments along with their results are presented in Sections 4.3 and 4.4 to evaluate the capability of this framework to predict the workload, power consumption and estimate the total cost for different VMs at the operation of Cloud services.

4.2 Energy-based Cost Prediction Framework

The cost mechanisms that are offered by Cloud service providers have become sophisticated, as customers are charged per month, hour, minute or second based on the resources they utilise. Nevertheless, there are still limited, as customers are charged based on pre-defined tariffs for the resources they utilise. These pre-defined tariffs do not consider the variable cost of energy [9], which is considered as one of the biggest operational cost factors by Cloud infrastructure providers. Consequently, estimating the cost of Cloud services including the energy consumption can help the service providers offer suitable services that meet their customers' requirements.

Therefore, an energy-based cost prediction framework that aims to predict the workload and power consumption, as well as estimate the total cost for a number of VMs during service operation is introduced. The VMs workload including vCPUs, memory, disk and network is firstly predicted. The predicted VM workload is then correlated to PM workload in order to estimate the PM power consumption, from which the predicted VMs power consumption would be based on. After that, the total cost of VMs is estimated based on their predicted workload and power consumption.

As depicted in Figure 4-1, the energy-based cost prediction framework is implemented within the cost modeller (introduced in Section 3.2) and includes five main steps to predict the VMs workload and power consumption, then estimate the total cost of VMs. To achieve this aim, the following steps are required.

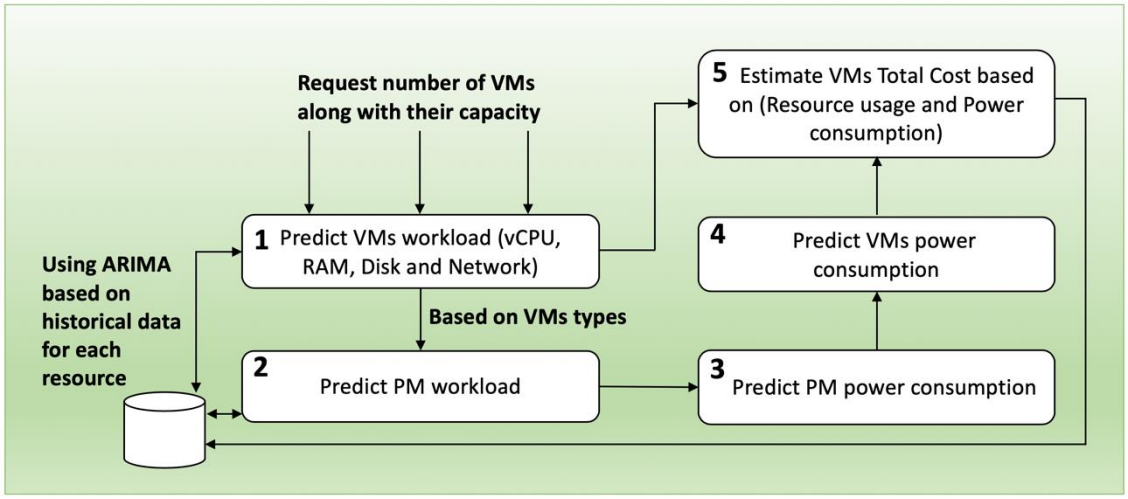


Figure 4-1: Energy-based Cost Prediction Framework.

4.2.1 VMs Workload Prediction

The first step of the framework is to predict VM workload for the next time interval, which is the requested number of VMs along with their capacity in terms of (vCPUs, memory, disk and network) to execute the application. Using the *Auto-Regressive Integrated Moving Average* (ARIMA) model, the VM workload is predicted based on historical workload patterns retrieved from a knowledge database. Cloud applications can experience different workload patterns based on the customers' usage behaviours, and these workload patterns consume power differently based on the resources they utilise.

As already pointed out, there are several workload patterns (discussed in Chapter 2, Section 2.3.1), such as static, *periodic*, *continuously changing*, *unpredicted*, and *once-in-a-lifetime*, as stated in [67]. The static workload pattern can be easily predicted, but there are many challenges that can obstruct the workload prediction when using other patterns. For example, other patterns may reflect temporary fluctuations of the workload such as continuously changing and once-in-a-lifetime or may be difficult to predict in advance such as the

unpredicted pattern. These patterns do not necessarily occur in data centres on a daily basis [101]. Therefore, it is essential to have approximated workload patterns that occur in the time series history to achieve a high prediction accuracy [101]. Thus, the periodic workloads can be more appropriate and precise to allow Cloud services to rapidly scale or descale the capacity to meet demand and dynamically control the cost of the infrastructure. Therefore, the simulated periodic workload pattern is considered for the historical data to be used in this framework.

The ARIMA model is a time series prediction model that has been used widely in different domains, including economics and finance, owing to its sophistication and accuracy [155]. ARIMA model is a generalisation of the *Auto-Regressive Moving Average* (ARMA) (p, q) model that contains two components: 1) *Auto-Regressive* (AR): the number of autoregressive parameters, which is indicated by (p), and 2) *Moving Average* (MA): the highest order of the moving average parameters represented by (q). In order to obtain the ARIMA model, ARMA models can be extended with an integrational (I) in order to convert a non-stationary time-series data to a stationary one, which is indicated by the differencing (d) value. Thus, the ARIMA model is generally stated as ARIMA (p, d, q), which consists of three main components: the order of autoregressive (p), the degree of differencing (d) and the order of moving average (q); further details about the ARIMA model can be found in [155].

A number of works, as in [101], [155], [156], have used the ARIMA model to predict workload in the cloud environment; though their objectives do not consider predicting the energy consumption and the total cost of VMs. Hence, the same approach using ARIMA model is applied in this thesis to predict the workload, but with the objectives of predicting the energy consumption and the total cost of VMs. Unlike other prediction methods such as sample average and single exponential smoothing, ARIMA is a powerful, quicker and more flexible model to predict time series data with low computational overhead [157], [101]. It takes multiple inputs as historical observations and outputs multiple future observations depicting the seasonal trend. Also, it can be used for seasonal or non-seasonal time-series data. The type of seasonal ARIMA model is used in this thesis as the targeted workload patterns are reoccurring and showing seasonality in time intervals. To use the ARIMA model for predicting the VM

workload, the historical time series workload data has to be stationary, otherwise, Box and Cox transformation [158] and data differencing methods are used to make these data series stationary. Further, the model selection of ARIMA can be automatically processed in the R package [159] using the (***auto.arima***) function, which selects the best fit model of ARIMA based on the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) value [155].

Once the VMs workload using the ARIMA model based on historical data is predicted, prediction of PMs workload and the PMs/VMs power consumption using regression models take place next.

4.2.2 PMs Workload Prediction

Once the VMs workload is predicted, the second step is to understand how this workload would be reflected on the physical resources and predict the PMs workload, which is based on PM CPU utilisation. This would require measuring the relationship between the number of vCPUs and the PM CPU utilisation for a PM. Therefore, the relationship between the number of vCPUs and the PMs' CPU utilisation is characterised for the targeted PMs. For the purpose of this framework, two different PMs (Host A and Host B, see Section 4.3.1) on the Cloud testbed have been characterised with regression models, as shown in Figures 4-2 and 4-3, respectively. These experiments were carried on the Cloud testbed by stressing the CPU to its full capacity using the *Stress-ng* tool [73], (see Section 4.4).

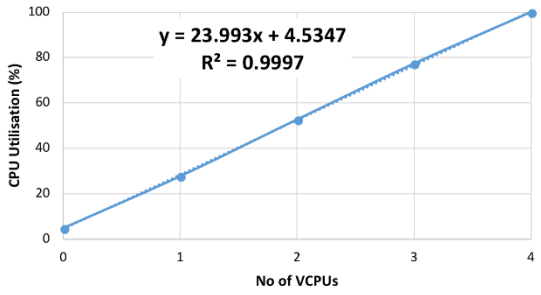


Figure 4-2: Number of vCPUs vs CPU Utilisation for Host A.

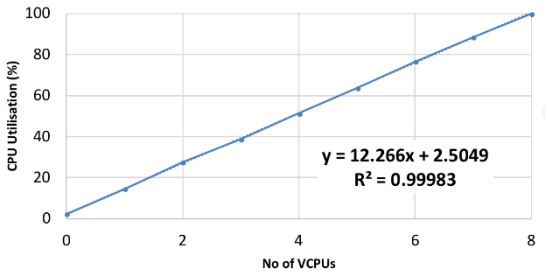


Figure 4-3: Number of vCPUs vs CPU Utilisation for Host B.

A linear regression model has been applied to predict the PMs CPU utilisation based on the used ratio of the requested number of vCPUs for the VMs

with consideration of its current workload as the PM may be running other VMs already [79], [92]. The following Equation is used (4.1):

$$\mathrm{PM}x_{PredUtil} = \left(\alpha \times \left(\sum_{y=1}^{VMCount} (\mathrm{VM}y_{ReqvCPUs} \times \frac{VMy_{PredUtil}}{100})\right) + \beta\right) + (PMx_{CurrUtil} - PMx_{IdleUtil}) \quad (4.1)$$

where $\mathrm{PM}x_{PredUtil}$ is the predicted PM CPU utilisation; α is the slope and β is the intercept of the CPU utilisation. The $\mathrm{VM}y_{ReqvCPUs}$ is the number of requested vCPUs for each VM and $VMy_{PredUtil}$ is the predicted utilisation for each VM. The $PMx_{CurrUtil}$ is the current PM utilisation and $PMx_{IdleUtil}$ is the idle PM utilisation.

4.2.3 PMs Power Consumption Prediction

After predicting the PMs workload, the third step is to predict the PMs power consumption based on the correlation of the predicted PM workload with PM power consumption. Thus, the considered PMs need to be characterised in terms of their power consumption in relation with CPU utilisation using regression models, as shown in Figures 4-4 and 4-5, respectively.

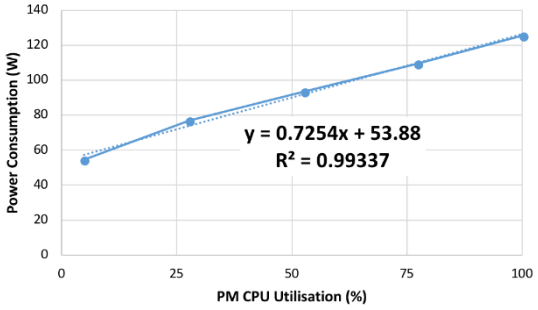


Figure 4-4: CPU Utilisation vs Power Consumption for Host A.

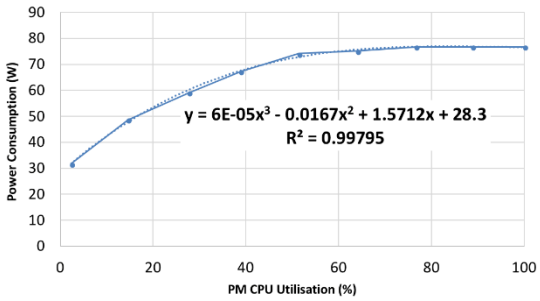


Figure 4-5: CPU Utilisation vs Power Consumption for Host B.

Therefore, the PMs predicted power consumption, $\mathrm{PM}x_{PredPwr}$ measured in Watts, can be identified using a linear relation with the predicted PMs CPU utilisation, as shown in Figure 4-4 and in Equation (4.2). α and β are the slope and interceptor values obtained from the regression relation, and $\mathrm{PM}x_{PredUtil}$ is predicted PM CPU utilisation.

$$\mathrm{PM}x_{PredPwr} = (\alpha \times (\mathrm{PM}x_{PredUtil}) + \beta) \quad (4.2)$$

However, not all existing PMs necessarily follow a linear power model in relation to their CPU utilisation, since the PMs are heterogeneous in nature, as shown for example in Figure 4-5. In this case, other regression models, such as polynomial, can be used to characterise the relation between the power consumption and CPU utilisation of the targeted PM, as shown in Equation (4.3).

$$\mathrm{PM}x_{PredPwr} = (\alpha(\mathrm{PM}x_{PredUtil})^3 - \gamma(\mathrm{PM}x_{PredUtil})^2 + \delta(\mathrm{PM}x_{PredUtil}) + \beta) \quad (4.3)$$

where α, γ and δ are all slopes, β is the intercept and $\mathrm{PM}x_{PredUtil}$ is predicted PM CPU utilisation.

4.2.4 VMs Power Consumption Prediction

The fourth step of this framework is to attribute the predicted PM power consumption to the newly requested VM and to the VMs already running on the physical host. Hence, the power consumption model of Equation (3.3) presented in Section 3.4 is used to predict power consumption for the new VM, $\mathrm{VM}x_{PredPwr}$, but with different notations, as shown in Equation (4.4):

$$\begin{aligned} \mathrm{VM}x_{PredPwr} = PMx_{IdlePwr} \times \left(\frac{\mathrm{VM}x_{ReqvCPUs}}{\sum_{y=1}^{VMcount} \mathrm{VM}y_{ReqvCPUs}} \right) \\ + (PMx_{PredPwr} - PMx_{IdlePwr}) \\ \times \left(\frac{\mathrm{VM}x_{(PredUtil \times ReqvCPUs)}}{\sum_{y=1}^{VMcount} \mathrm{VM}y_{(PredUtil \times ReqvCPUs)}} \right) \end{aligned} \quad (4.4)$$

where $\mathrm{VM}x_{PredPwr}$ is the predicted power consumption for one VM measured in Watt. The $PMx_{IdlePwr}$ is the idle power consumption and $PMx_{PredPwr}$ is the predicted power consumption for a single PM. $\mathrm{VM}x_{ReqvCPUs}$ is the requested number of vCPUs and $\sum_{y=1}^{VMcount} \mathrm{VM}y_{ReqvCPUs}$ is the total number of vCPUs for all VMs on the same PM. $\mathrm{VM}x_{(PredUtil \times ReqvCPUs)}$ is the predicted VM CPU utilisation times the number of vCPUs assigned to the given VM; and $\sum_{y=1}^{VMcount} \mathrm{VM}y_{(PredUtil \times ReqvCPUs)}$ is the predicted VMs CPU utilisation times the number of vCPUs for a set of VMs hosted by the same PM.

4.2.5 VMs Total Cost Estimation

The final step in this framework is to estimate the total cost of the VM based on the validated results from the predicted VM workload (Section 4.2.1) and the predicted VM power consumption (Section 4.2.4). The energy providers usually charge by kWh. Therefore, the conversion of the predicted power consumption to energy is required using Equation (3.5) presented in Section 3.4, but with substitution of $\mathrm{VM}x_{Energy}$ with $\mathrm{VM}x_{PredEnergy}$ and $\mathrm{VM}x_{Pwr}$ with $\mathrm{VM}x_{PredPwr}$, as shown in Equation (4.5):

$$\mathrm{VM}x_{PredEnergy} = \frac{\mathrm{VM}x_{PredPwr}}{1000} \times \frac{Time_s}{3600} \tag{4.5}$$

To estimate the total cost for the VM, Equation (3.6) presented in Section 3.4 is used with different notations, as shown in Equation (4.6):

$$\begin{aligned} \mathrm{VM}x_{EstTotalCost} = & \left(\left(\mathrm{VM}x_{ReqvCPUs} \times \frac{VMx_{PredUtil}}{100}\right) \times (Cost\ per\ vCPU\ \times Time_s)\right) \\ & + \left(\mathrm{VM}x_{PredRAMUsage} \times (Cost\ per\ GB\ \times Time_s)\right) \\ & + \left(\mathrm{VM}x_{PredDiskUsage} \times (Cost\ per\ GB\ \times Time_s)\right) \\ & + \left(\mathrm{VM}x_{PredNetUsage} \times (Cost\ per\ GB\ \times Time_s)\right) \\ & + \left(\mathrm{VM}x_{PredEnergy} \times Cost\ per\ kWh\right) \end{aligned} \tag{4.6}$$

where $\mathrm{VM}x_{EstTotalCost}$ is the estimated total cost of the VM. The $\mathrm{VM}x_{ReqvCPUs}$ is the number of requested vCPUs for each VM and $\mathrm{VM}x_{PredUtil}$ is the predicted utilisation for each VM times the cost for the requested vCPUs for a period of time, $Time_s$. $\mathrm{VM}x_{PredRAMUsage}$, $\mathrm{VM}x_{PredDiskUsage}$ and $\mathrm{VM}x_{PredNetUsage}$ denoted the predicted resource usage of memory, disk and network, respectively, times the cost of each resource for a period of time. $\mathrm{VM}x_{PredEnergy}$ is the predicted energy consumption of the VM times the energy cost as considered by the energy providers.

4.3 Implementation

The energy-based cost prediction framework is introduced in this research to estimate the total cost of the VMs during service operation based on historical

workload patterns. Thus, in order to evaluate this framework, a number of experiments have been conducted on the Cloud testbed (see Section 4.4.2) to synthetically generate historical workload data. The historical data has been generated to represent real workload patterns of Cloud applications (discussed in Section 4.2.1), including a periodic workload, by stressing all the resources (CPU, memory, disk and network) on different types of VMs using the *Stress-ng* tool [73] (see Section 4.4.1). The prediction process works by firstly predicting the VM workload using the (***auto.arima***) function in R package [159] to automatically select the best fit model of ARIMA based on AIC or BIC value. The process is then going through the steps of the introduced framework to consider the correlation between the physical and virtual resources in order to predict the power consumption of the VMs. Finally, the total cost of the VMs when being run on different PMs is estimated based on their predicted workload and power consumption.

4.3.1 Characterisation of Physical Machines

Two different PMs on the 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 an eight-core E3-1230 V2 Intel Xeon CPU. Also, each PM has a Watt meter [143] attached to directly measure the power consumption. Heterogeneous VMs are created and their monitoring is performed through Zabbix [150], which is also used for resource usage monitoring.

4.4 Experiments and Evaluation

4.4.1 Design of Experiments

The overall aim of the experiments is to demonstrate that the energy-based cost prediction framework is capable of predicting the workload, power consumption as well as estimating the total cost of heterogeneous VMs when being run on heterogeneous PMs.

Three direct experiments have been conducted by using three types of VMs with the objective to 1) understand the relation between each PM's CPU utilisation and the number of vCPUs, 2) understand the power characteristics of

each PM with their CPU utilisation, and 3) predict the workload, power consumption and estimate the total cost for a number of VMs when being run on different PMs.

To design the experiments, historical periodic workload patterns have been generated synthetically to represent real workload patterns of Cloud applications by conducting a number of experiments to stress all the resources (CPU, memory, disk and network) on different types of VMs, as discussed in Section 4.3.

In order to generate a periodic workload pattern for each VM type, a time interval of four slots (30 minutes each) is considered. The first three intervals (slots) are used as the historical data set for prediction, and the last interval (slot) is used as the testing data set to evaluate the predicted results. A similar approach is used in [160] and followed in this thesis.

Note that each experiment is set to run for 30 minutes in order to ensure it runs long enough and produce relevant data to simulate the periodic workload patterns. Further, in order to use the ARIMA model, it would require having at least 50 but preferably more than 100 observations to train the model [155]. Therefore, all experiments have four-time intervals 30 minutes each, equal to 120 minutes (observations), and thus applied to the experiments conducted in this Chapter and subsequent Chapters 5 and 6 to ensure consistency.

4.4.2 Evaluation

The conducted experiment shows the prediction results for three types of VMs, small, medium and large on two different PMs, (Host A and Host B), having different characteristics. The aim of this experiment is to evaluate the capability of the proposed framework to predict the workload, power consumption and estimate the total cost for a number of VMs with different workload when being run on different PMs.

In terms of the historical and testing data sets, Figures 4-6, 4-7 and 4-8 depict the results of the predicted versus the actual VMs workload, including (CPU, RAM, disk, and network) usage for the VMs. Despite the periodic utilisation peaks, the predicted VMs' CPU and RAM workload results closely

match the actual results, which shows the strength of the ARIMA model for predicting based on historical seasonal data, repeated patterns of the periodic workload and give a very accurate prediction accordingly. The predicted VMs' disk and network workload also match the actual workload, but with less accuracy as compared to the CPU and RAM prediction results. This can be justified because of the high variations in the generated historical periodic workload pattern of the disk and network not closely matching in each interval, whereas the generated historical periodic workload patterns for the CPU and RAM usage are closely matched in each interval. Besides the predicted VMs' workload mean values, the results also show the high and low 95% and 80% confidence intervals for the predicted workload of each VM based on the ARIMA model.

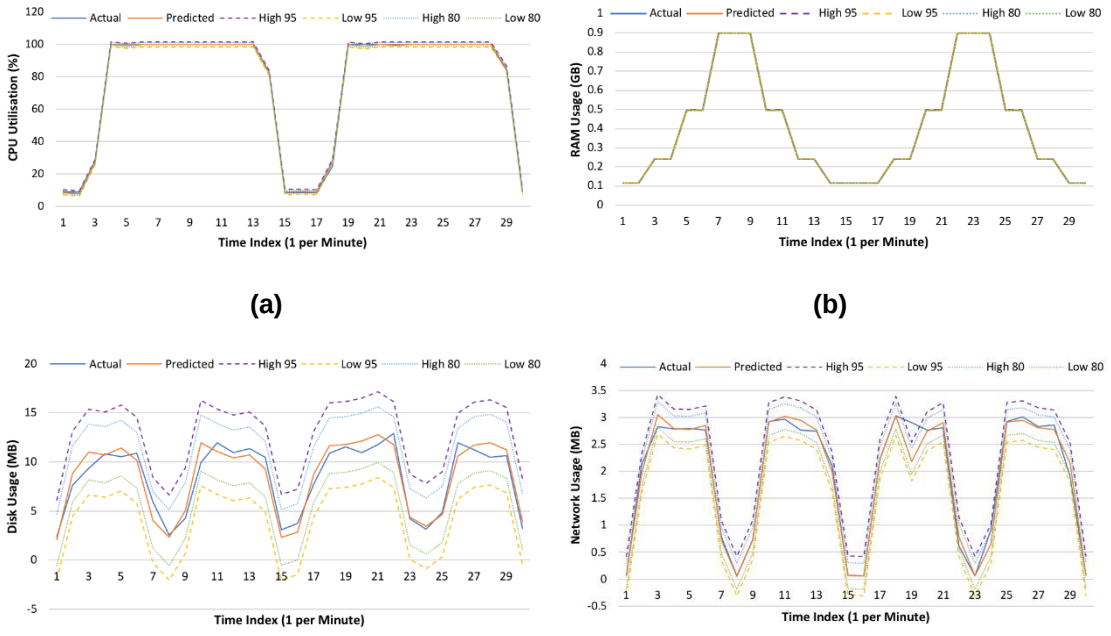


Figure 4-6: The Workload Prediction for Small VM (for 30 minutes).

Table 4-1: Prediction Accuracy for Small VM.

Parameters	ME	RMSE	MAE	MPE	MAPE
CPU Utilisation	0.057922	0.638338	0.282995	0.176069	1.324204
RAM Usage	0.000060	0.000115	0.000072	0.015359	0.018484
Disk Usage	0.1188962	0.975295	0.841385	-1.49987	12.05513
Network Usage	-0.015988	0.167085	0.089504	-2.02527	5.942

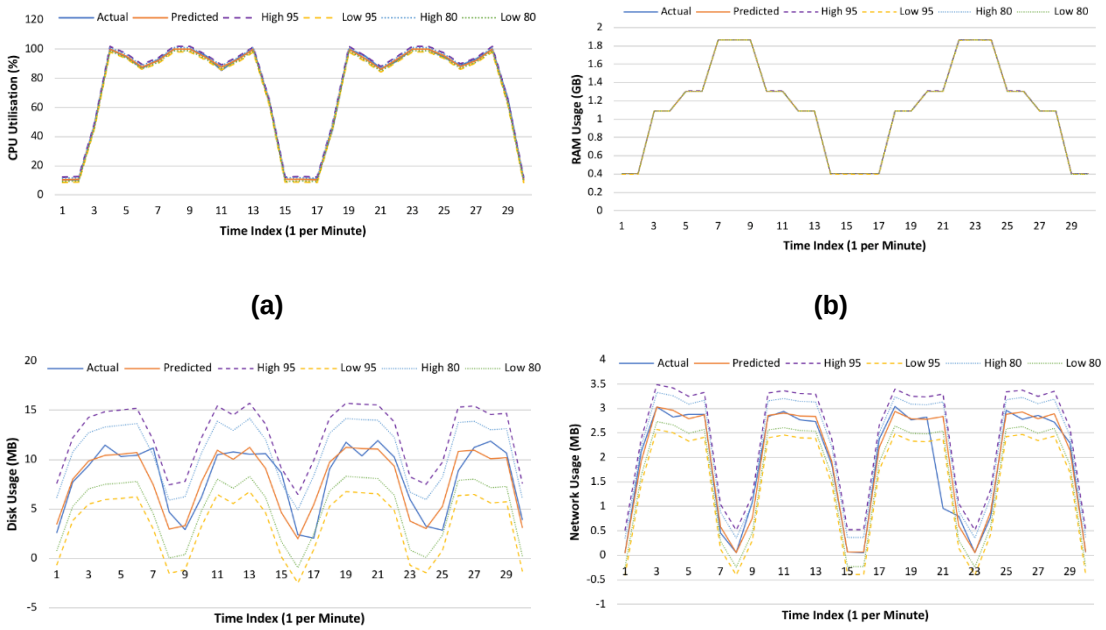


Figure 4-7: The Workload Prediction for Medium VM (for 30 minutes).

Table 4-2: Prediction Accuracy for Medium VM.

Parameters	ME	RMSE	MAE	MPE	MAPE
CPU Utilisation	-0.0592	0.93451	0.631026	-0.15173	1.07904
RAM Usage	0.000003	0.00029	0.000193	0.004484	0.018850
Disk Usage	-0.22049	1.57393	1.189894	-5.46191	20.25606
Network Usage	0.046753	0.36071	0.145621	1.11802	9.038321

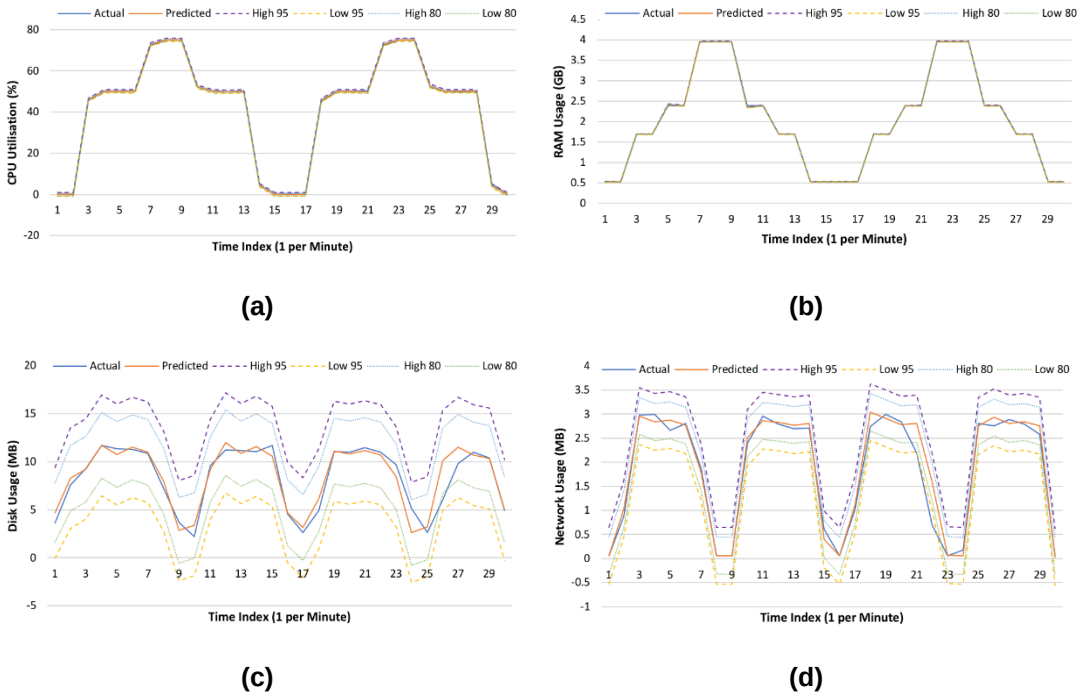


Figure 4-8: The Workload Prediction for Large VM (for 30 minutes).

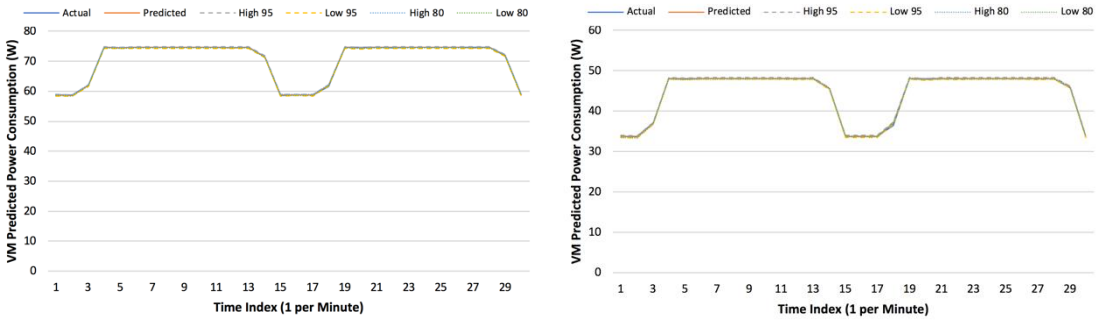
Table 4-3: Prediction Accuracy for Large VM.

Parameters	ME	RMSE	MAE	MPE	MAPE
CPU Utilisation	0.03765	0.299769	0.137823	0.309809	6.615192
RAM Usage	0.000004	0.008671	0.002587	-0.00675	0.107601
Disk Usage	0.1838898	1.116114	0.733408	0.924781	12.64005
Network Usage	0.0657477	0.225631	0.132185	-6.13982	17.56377

In terms of prediction accuracy, a number of metrics have been used to evaluate the predicted workload (CPU, RAM, disk and network) for small, medium and large VMs based on periodic workload patterns as presented in Tables 4-1, 4-2 and 4-3, respectively. These metrics include, *Mean Error (ME)* which measures the average error of the predicted values; *Root Mean Squared Error (RMSE)* which depicts the square root of the variance measured by the mean absolute error; *Mean Absolute Error (MAE)* is the average of the absolute value of the difference between predicted value and the actual value; *Mean Percentage Error (MPE)* is the computed average of percentage errors by which the predicted values vary from the actual values; and *Mean Absolute Percent Error (MAPE)* is the average of the absolute value of the difference between the predicted value and the actual value explained as a percentage of the actual

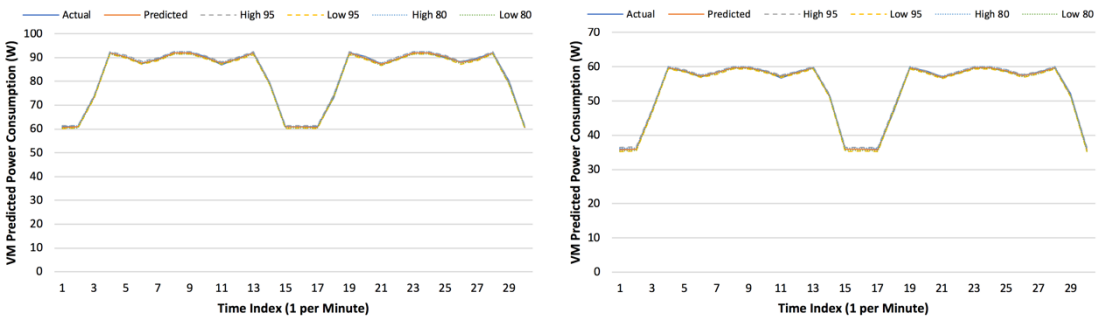
value [161]. When the values of these metrics are too low or close to zero, it indicates that the prediction method has achieved very high prediction accuracy.

Based on the predicted workload for each VM, their power consumption is predicted through the remaining steps within the framework. Figures 4-9, 4-10 and 4-11 show the predicted versus the actual results of the power consumption for small, medium and large VMs when being run on Host A and Host B, noting that Host B is more energy efficient as compared to Host A. Also, the predicted power consumption attribution for each VM is affected by the variation in the predicted CPU utilisation of all VMs, hence the predicted power consumption of all VMs is closely matched the pattern of the predicted VMs CPU utilisation, as shown in Figures 4-6, 4-7 and 4-8. In terms of prediction accuracy, a number of metrics have been used to evaluate the predicted power consumption for small, medium and large VMs based on periodic workload patterns as presented in Table 4-4.



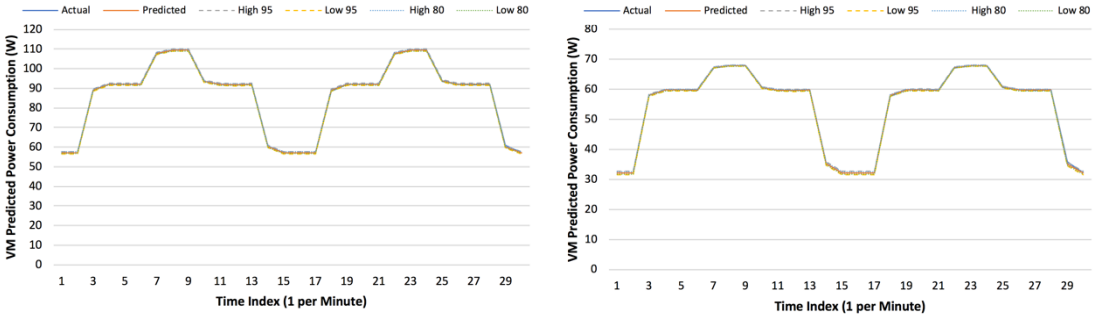
(Host A) **(Host B)**

Figure 4-9: Predicted Small VM Power Consumption.



(Host A) **(Host B)**

Figure 4-10: Predicted Medium VM Power Consumption.



(Host A) **(Host B)**

Figure 4-11: Predicted Large VM Power Consumption.

Table 4-4: Prediction Accuracy for The Predicted Power Consumption for all VMs on (Host A and Host B).

Parameter	VMs	Hosts	ME	RMSE	MAE	MPE	MAPE
VMs Power Consumption	Small VM	Host A	0.01049666	0.1050482	0.04551519	0.02957622	0.1178556
		Host B	0.01007936	0.1110991	0.04925524	0.01709142	0.0759957
	Medium VM	Host A	-0.0124435	0.2178242	0.1455701	-0.0238362	0.2762649
		Host B	-0.02060695	0.3252957	0.2196544	-0.0242161	0.2635127
	Large VM	Host A	0.02621121	0.2086939	0.09594997	0.01031316	0.1175031
		Host B	0.00013134	0.1663338	0.06292857	-0.0310118	0.1377464

This framework is also capable of estimating the total cost for a number of VMs hosted/running on different PMs as shown in Figure 4-12, which presents the estimated total cost of small, medium and large VMs running on different PMs (Host A and Host B).

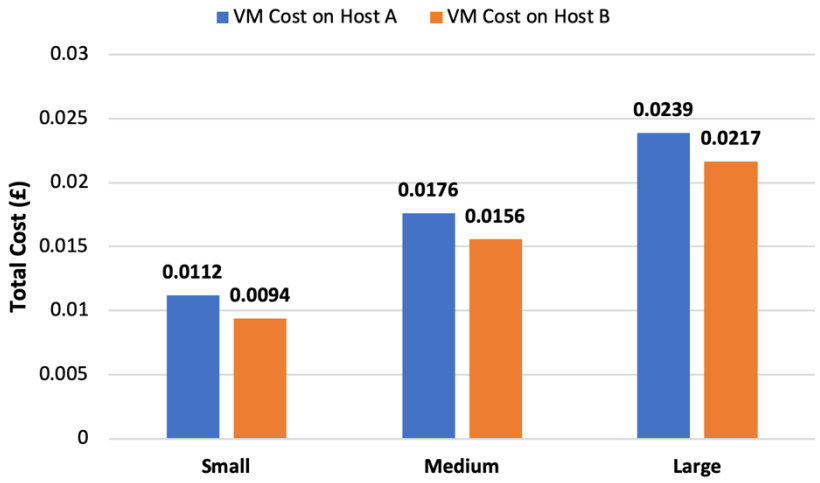


Figure 4-12: The Estimated VMs Total Cost on Host A and Host B.

In addition, Figure 4-12 shows the estimated cost for a number of VMs running on different PMs. As the VMs are heterogeneous, the costs of VMs are therefore different. The cost of small VM is about two times smaller than medium VM and three times smaller than large VM, which is based on their resource usage and energy consumption by each VM. The predicted energy efficiency of Host B plays an important role in reducing the total cost of the VMs comparing to Host A. As a result, choosing the more energy efficient host (Host B) to run the VMs can achieve 16.25% cost-saving for the small VM, 11.75% for the medium VM and 9.38% for the large one, as shown in Figure 4-13.

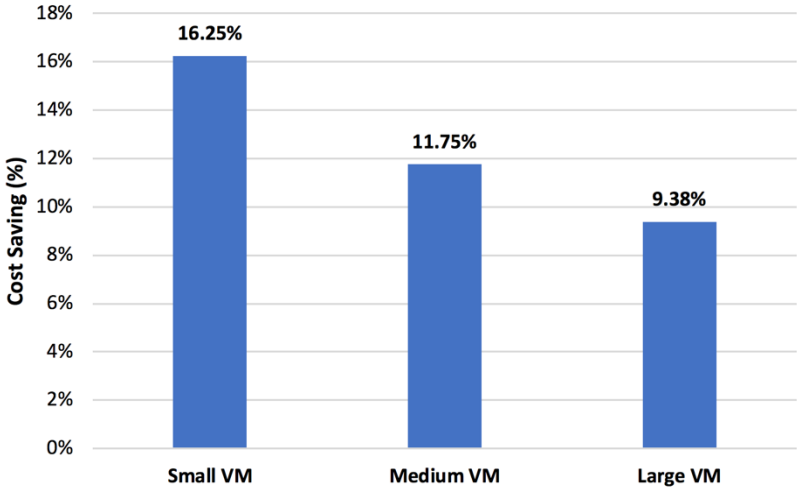


Figure 4-13: The Estimated VMs Cost Saving on Host B.

Despite the combination of different types of VMs with different workloads running on different PMs, the accuracy metrics indicate that the predicted VMs workload and power consumption achieve high prediction accuracy along with the estimated total cost.

4.5 Summary

The proposed energy-based cost prediction framework for predicting resource usage, power consumption and estimating the total cost of heterogeneous VMs during service operation has been presented and discussed comprehensively in this chapter. The chapter further has been followed by a demonstration of a number of experiments along with their results to evaluate the capability of the proposed framework for predicting the workload, power consumption and

estimating the total cost of VMs based on historical workload pattern when being run on heterogeneous PMs.

More care has been put into the overall process from experiment design, implementation, data collection, data analysis to ensure it is thorough and consistent. For this purpose, statistical analyses (linear and polynomial regressions) have been used to compare and validate the model output with the real system output, and the values of R-squared have been given accordingly. In terms of prediction, the first three intervals have been used as the training data set for prediction, and the last interval has been used as the testing data set to evaluate the predicted results. The predicted versus the actual VMs workload (CPU, RAM, disk, and network) along with their power consumption have been validated using five accuracy metrics, and the high and low 95% and 80% confidence intervals have also used. Thus, considering all these methods have helped the proposed model validation and gave confidence in its results obtained.

Chapter 5. Performance and Energy-based Cost Prediction Framework

5.1 Overview

In this chapter, a performance and energy-based cost prediction framework that aims to estimate the total cost of VMs by considering their resource usage and power consumption, while maintaining the expected level of application performance is presented in Section 5.2. This framework includes two approaches that can be used for VMs consolidation and resource provisioning in order to design a cost-effective strategy and prevent performance loss at different levels. A number of experiments along with their results to evaluate the capability of the proposed framework to estimate live migration and auto-scaling total cost for heterogeneous VMs at service operation are presented in Sections 5.3 and 5.4.

5.2 Performance and Energy-based Cost Prediction Framework

The cost mechanisms that are employed by different Cloud service providers significantly influence the adoption of Cloud Computing within the IT industry. With the increasing cost of electricity, Cloud providers consider energy consumption as one of the biggest operational cost factors to be managed within their infrastructures. Most of the existing studies have focused on minimising the energy consumption and maximising the resource usage, instead of improving the performance [121]. Further, Cloud providers such as Amazon [22], have established their SLAs based on service availability without such an assurance of the service performance. For instance, during service operation, consider the situation where a number of VMs are running on the same PM, and each VM is allocated its fair share of resources. If the VM's workload increases (stretching its capacity to its limits) and no resources are available to handle that increment (e.g., the workload exceeds the acceptable level of CPU such as 95% threshold), resource competition may occur leading to VMs' performance degradation [121], which may affect the fulfilment of the SLAs and hence the cloud provider's

revenue. Hence, to prevent such performance loss, it is necessary to have preventive actions such as VMs re-allocation through live migration and VMs auto-scaling. Therefore, a performance and energy-based cost prediction framework that supports the potential actuators (e.g., migrating and auto-scaling VMs) to handle the performance variation in a cost-efficient manner is proposed. This framework aims towards predicting PMs/VMs workload and power consumption as well as estimating the total cost of the VMs incurred by live migration and auto-scaling. Thus, the energy-based cost prediction framework (discussed in Chapter 4) is used in this Chapter.

Generally, the performance and energy-based cost prediction framework implemented inside the cost modeller (introduced in Section 3.2) can be described using a classic MAPE (Monitor, Analyse, Plan, Execute) control loop [162], as illustrated in Figure 5-1.

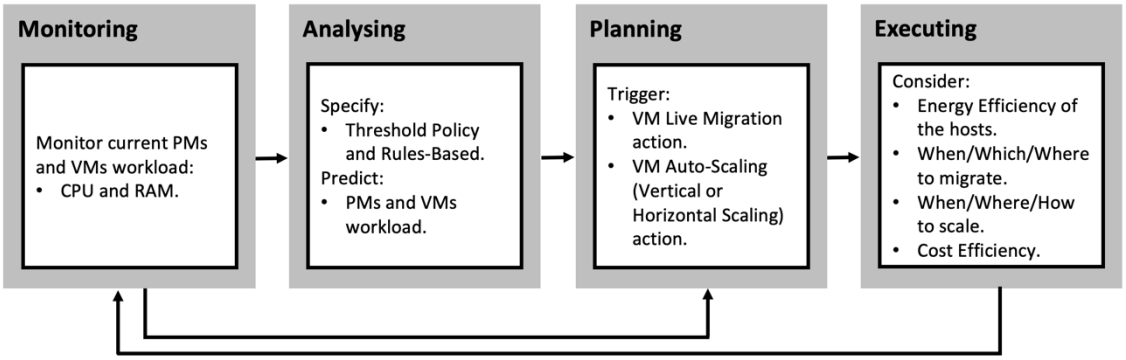


Figure 5-1: Performance and Energy-based Cost Prediction Framework.

A brief explanation of each phase is provided as follows:

- **Monitoring**: the PMs and VMs workload (CPU utilisation and RAM usage) are continuously monitored and the data are collected through a monitoring system [150].
- **Analysing**: the collected data are analysed and threshold policies and rules-based are set in order to identify any changes in the behaviour of the workload. During this phase, the framework determines whether it is necessary to predict the workload for the next time interval based on the threshold policy and the rule-based.

- **Planning**: based on the output of the analysis phase, a proper action (e.g., VM live migration or auto-scaling) is selected and the target component (**Execute**) is informed to start the execution of the action.
- **Executing**: the energy efficiency of the hosts is considered as a key factor, which influences the overall cost of the performed action in this phase. Moreover, this phase finds when to migrate, which VMs to migrate and where to migrate. Also, it decides when to scale, how to scale the VMs and where to scale.

Further details of this framework and the role of live migration and auto-scaling are presented next in Section 5.2.1 and Section 5.2.2, respectively.

5.2.1 VMs Live Migration Prediction Models

VMs live migration is an important mechanism to improve resource utilisation and achieve energy efficiency in Clouds. Live migration allows VMs to move from one PM to another without any interruption in the service. This mechanism plays an important role in load balancing among the PMs and reduces the overall energy consumption [114]. However, VMs live migration is a resource-intensive operation [99], which has an impact on the performance of the migrating VM as well as the services running on other VMs [12], [13], [100], [118], [124]. Besides, there are additional costs [14] in terms of migration time and energy overhead that need further consideration [15], [16]. Hence, understanding the impact of VM live migration is essential to design an effective consolidation strategy.

Previous studies show that in most situations, live migration overhead is acceptable but cannot be ignored as stated in [112], [163], [113]. As a result, live migration overhead needs to be taken into account when the migration decision is about to be made [129], [130]. Consequently, estimating the future cost of Cloud services can help the service providers offer suitable services that meet their customers' requirements. Thus, a proactive framework has the advantage of taking preventive actions (e.g., re-allocating and migrating VMs) at an early stage to avoid service performance degradation. The effectiveness of such framework depends on potential actuators/decisions to detect the overloaded hosts in order to decide when to migrate, which VMs to migrate and where to migrate.

The proposed framework is implemented inside the cost modeller (introduced in Section 3.2) and supports decision-making regarding live migration cost while at the same time being aware of the impact on other quality characteristics such as energy consumption and performance of the application. This framework is aimed towards predicting PMs/VMs workload and power consumption as well as estimating the total cost and the recovery cost of the VMs incurred by live migration, as depicted in Figure 5-2.

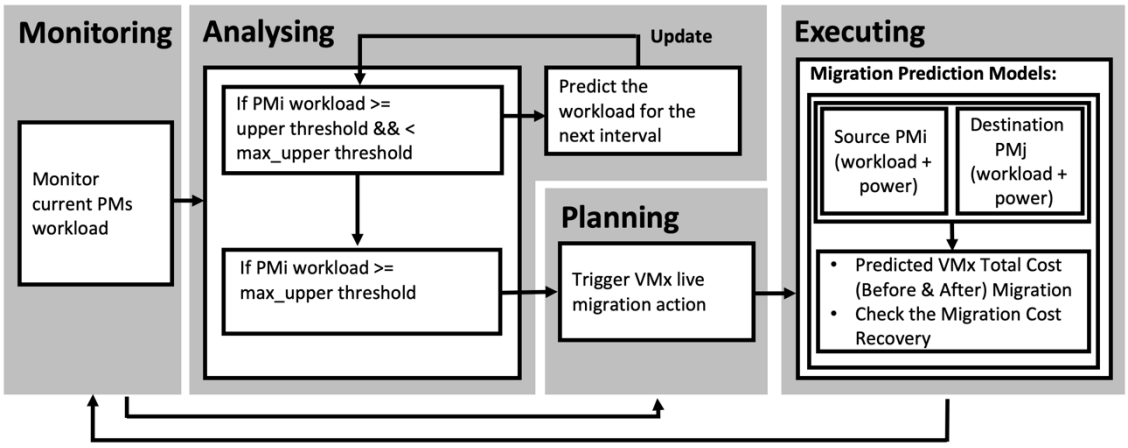


Figure 5-2: Performance and Energy-based Cost Prediction Framework (Live Migration).

To achieve this aim, several steps are required in order to predict the PMs/VMs workload and power consumption, then estimate the total cost of the migrated VMs as explained next.

Step 1: The PMs CPU utilisation and RAM usage (upper and max_upper) thresholds (e.g., 85% and 95%) are set and the source PM*i* workload is monitored periodically. If the PM*i* workload equals or exceeds the max_upper threshold (e.g., 95%), VM live migration is performed as described in Algorithm 5.1, using the *pre-copy* VM live migration technique (introduced in Section 2.7.1). The list of the algorithms parameters and their notations is shown in Table 5-1.

Step 2: If the PM*i* workload is in the range of [upper and max_upper threshold], then predict the PM*i* workload for the next time interval (e.g., every 5 minutes [164]) using the ARIMA model based on historical workload patterns (as introduced in Chapter 4). This prediction helps detect the workload to control the number of migrations in order to avoid unnecessary migration caused by the small peaks in the workload (false alarm). If the predicted workload for the next

interval exceeds the max_upper threshold, VM live migration is performed as described in Algorithm 5.1.

Table 5-1: Summary of Notations.

Notation	Description
PM*i*	the source physical machine
PM*j*	the destination physical machine
VM*x*	the candidate VM to migrate / the overloaded VM to scale
C_CPU_PM	total CPU capacity of the PM
C_RAM_PM	total memory capacity of the PM
U_CPU_PM	used CPU capacity of the PM ($\sum_{y=1}^{VM\_Count}(\mathrm{vCPU})$)
U_RAM_PM	used memory capacity of the PM ($\sum_{y=1}^{VM\_Count}(\mathrm{RAM})$)
C_CPU_VM	total CPU capacity of the VM
C_RAM_VM	total memory capacity of the VM
U_CPU_VM	used CPU capacity of the VM
U_RAM_VM	used memory capacity of the VM
I_CPU_VM	increment CPU capacity of the VM
I_RAM_VM	increment memory capacity of the VM

Algorithm 5.1: Performance Prediction

Initialise: PM workload = $\left(\frac{\mathrm{U\_CPU\_PM}}{\mathrm{C\_CPU\_PM}}, \frac{\mathrm{U\_RAM\_PM}}{\mathrm{C\_RAM\_PM}}\right)$;

PM max_upper threshold = 0.95 × (C_CPU_ PM, C_RAM_ PM);

PM upper threshold = 0.85 × (C_CPU_ PM, C_RAM_ PM);

Predicted workload = null.

Input: PMs list.

```
1: for each (PMi in PMs list) do
2:     if (PMi workload ≥ PMi max_upper threshold) then
3:         perform VM live migration using (Algorithm 5.2);
4:         break
5:         else
6:             if (PMi workload ≥ PMi upper threshold) && (PMi workload < PMi max_upper threshold) then
7:                 Predicted workload ⟵ predict the (PMi workload) for the next interval using the ARIMA model.
8:                 PMi workload = Predicted workload;
9:             end if
10:    end if
11: end for
```

Step 3: The proposed Algorithm 5.2 is used to identify the candidate VM*x* to be migrated and appropriate destination PM*j* to host it. This algorithm combines live migration with re-allocation in order to minimise the overall cost of migration by re-allocation the VMs to the most energy efficient host, if possible. To do so, the PMs are ranked in decreasing order according to their energy efficiency, whereas the VMs are ranked in increasing order of their workload. The energy efficiency of the hosts (source PM*i* and destination PM*j*) is computed based on Equation (3.3) presented in Section 3.4. Thus, the energy efficiency of

the hosts can be given by: PM power = $\frac{\text{PM}i \text{ (power of the source)}}{\text{PM}j \text{ (power of the destination)}}$, whereas; $\text{PM}i \text{ (power of the source)} = \sum_{y=1}^{VMcount} \text{VM}pwr$ and $\text{PM}j \text{ (power of the destination)} = \sum_{y=1}^{VMcount} \text{VM}pwr$. The $\sum_{y=1}^{VMcount} \text{VM}pwr$ denotes the sum of the VMs power consumption that are already running on the host, which includes the idle and active power consumption of the host. For example, if the PM power > 1, the destination host is more energy efficient than the source; if the PM power = 1, the destination host is similar to the source in terms of the energy efficient and if the PM power < 1, the destination host is less energy efficient than the source. Starting with the PM*j* with the lowest idle power (the most energy efficient host), and check if PM*j* has enough resources to meet the migration requirements while at the same time making sure that the destination host PM*j* will not exceed the upper threshold for allocating of the migrated VM*x*. The task is to select a matching candidate VM*x* for migration, considering firstly the one with the smallest workload. This ensures 1) the candidate VM*x* does not overload the destination PM*j*, 2) the source PM*i* workload decreases significantly once the migration has taken place, and 3) potentially increase the ability for recovering the migration costs.

Algorithm 5.2: VM Selection for Migration and PM Allocation

Initialise: VM workload = $\left(\frac{\text{U\_CPU\_VM}}{\text{C\_CPU\_VM}}, \frac{\text{U\_RAM\_VM}}{\text{C\_RAM\_VM}}\right)$;

PM workload = $\left(\frac{\text{U\_CPU\_PM}}{\text{C\_CPU\_PM}}, \frac{\text{U\_RAM\_PM}}{\text{C\_RAM\_PM}}\right)$;

PM upper threshold = 0.85 × (C_CPU_ PM, C_RAM_ PM);

PM power = $\frac{\text{PM}i \text{ (power of the source)}}{\text{PM}j \text{ (power of the destination)}}$; // to check the energy efficiency

Destination PM = null, Candidate VM = null.

Input: PMs list, VMs list.

Output: Candidate VM, Destination PM.

```
1: Sort the PMs list in decreasing order of the PM power;
2: Sort the VMs list on PMi in increasing order of the workload; // (on the source host)
3:    for each (VMx in VMs list) do
4:       for each (PMj in PMs list) do
5:          if ((PMj workload + VMx workload) < PMj upper threshold) then
6:             Destination PM = PMj;
7:             Candidate VM = VMx;
8:             break
9:          end if
10:      end for
11:   end for
12: return Candidate VM, Destination PM.
```

After identifying the candidate VM*x* and the destination PM*j*, the ARIMA model is used to predict the candidate VM*x* workload including (CPU, memory, disk and network) and identify the best fit model (as introduced in Section 4.2.1). Once the candidate VMx workload is predicted using the ARIMA model based on historical data, the next step is to predict the PMs (source and destination) workload and PMs/VMx power consumption using regression models.

Step 4: To predict the PMs workload represented as (PMs CPU utilisation), would require measuring the relationship between the number of vCPU and the PM CPU utilisation for the PMs, as shown in Figures 5-3, 5-4 and 5-5.

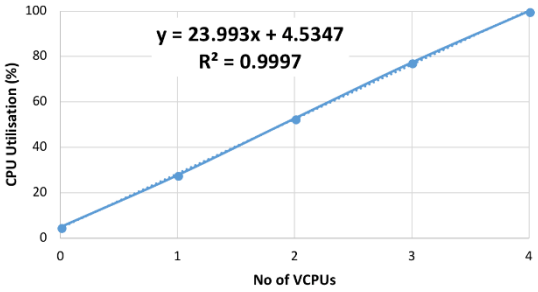


Figure 5-3: Number of vCPUs (VMx) vs PM CPU Utilisation (Source PMi), Host A.

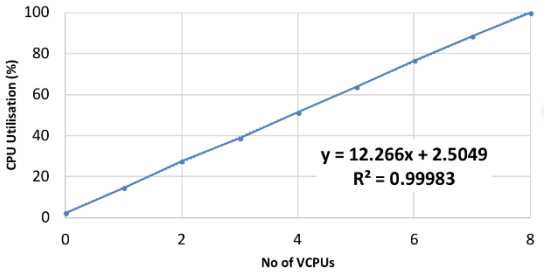


Figure 5-4: Number of vCPUs (VMx) vs PM CPU Utilisation (Destination PMj - most energy efficient PM), Host B.

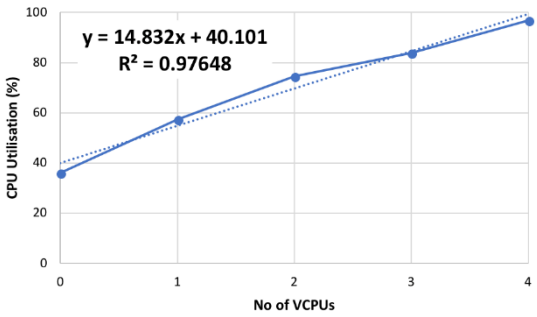


Figure 5-5: Number of vCPUs (VMx) vs PM CPU Utilisation (Destination PMj - less energy efficient PM), Host D.

The linear regression model of Equation (4.1) presented in Section 4.2.2 is used to predict the PMs CPU utilisation.

Step 5: The PMs power consumption is predicted based on the relationship between the predicted PM workload (PM CPU utilisation) with PM

power consumption on the PMs. Using a regression analysis, the relation is best described as linear regression for this particular PM*i*, as shown in Figure 5-6.

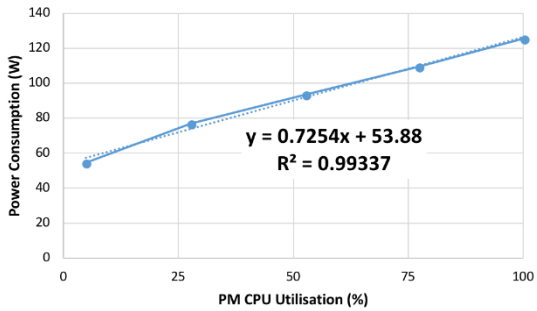


Figure 5-6: The PM CPU Utilisation vs Power Consumption (Source PMi), Host A.

The linear regression model of Equation (4.2) presented in Section 4.2.3 is used to predict the PMs power consumption.

As discussed in Chapter 4, not all existing PMs necessarily follow a linear power model in relation to their CPU utilisation, as shown in Figures 5-7 and 5-8.

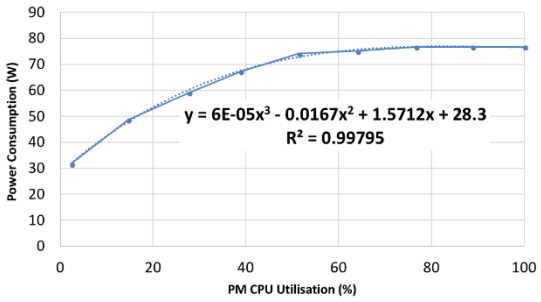


Figure 5-7: The PM CPU Utilisation vs Power Consumption (Destination PMj - most energy efficient PM), Host B.

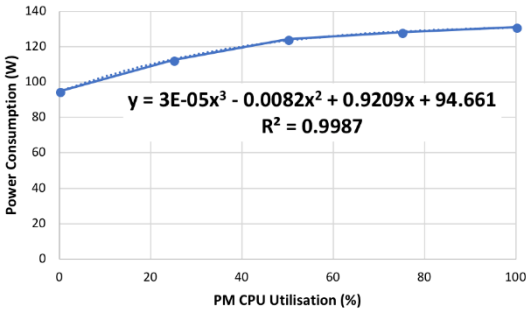


Figure 5-8: The PM CPU Utilisation vs Power Consumption (Destination PMj - less energy efficient PM), Host D.

In this case, other regression models, such as polynomial, can be used to characterise the relation between the power consumption and CPU utilisation of the targeted PMs, as presented in Equation (4.3), Section 4.2.3.

Step 6: The proposed Equation (4.4), in Section 4.2.4, is used to predict the VM*x* power consumption on both PMs (source and destination), then the conversion of the power consumption to energy is required using the Equation (4.5).

Step 7: This step estimates the total cost for the migrated VM*x* based on the predicted VM*x* resource usage in *Step 3* and the predicted VM*x* energy consumption in *Step 6* within this framework.

The total time required for migrating VM*x* can be given by:

$$T_{Mig} = (T_{Mig\_End} - T_{Mig\_Start}) \tag{5.1}$$

$$T_{Run\_Sou} = (T_{Run\_Sou\_Bef\_Mig} + T_{Mig}) \tag{5.2}$$

$$T_{Run\_Des} = (T_{Run\_Des\_Aft\_Mig} + T_{Mig}) \tag{5.3}$$

where T_{Mig} is the VM*x* total migration time measured by seconds. T_{Mig_Start} is the time when the migration is started and T_{Mig_End} is the time when the migration is ended. T_{Run_Sou} is the running time of the VM*x* on the PM*i* before migration starts plus the migration time T_{Mig} itself and $T_{Run\_Sou\_Bef\_Mig}$ is the running time of VM*x* before migration. T_{Run_Des} is the running time of the VM*x* on the PM*j* during and after migration and $T_{Run\_Des\_Aft\_Mig}$ is the running time of VM*x* after migration.

The total cost for VM*x* would require an estimation of the cost of the VM before and after the migration process. Hence, to estimate the total cost for VM*x* before migration, Equation (4.6) presented in Section 4.2.5 is used, but with different notations, as shown in Equation (5.4):

$$\begin{aligned} \mathrm{VM}x_{Est\_Cost\_PMi} = & \left(\left(\mathrm{VM}x_{ReqvCPUs\_PMi} \times \frac{VMx_{Pred\_U\_PMi}}{100} \right) \right. \\ & \left. \times \left(Cost\_vCPU \times T_{Run\_Sou} \right) \right) \\ & + \left(\mathrm{VM}x_{Pred\_R\_U\_PMi} \times \left(Cost\_GB \times T_{Run\_Sou} \right) \right) \\ & + \left(\mathrm{VM}x_{Pred\_D\_U\_PMi} \times \left(Cost\_GB \times T_{Run\_Sou} \right) \right) \\ & + \left(\mathrm{VM}x_{Pred\_N\_U\_PMi} \times \left(Cost\_GB \times T_{Run\_Sou} \right) \right) \\ & + \left(\mathrm{VM}x_{Pred\_E\_PMi} \times Cost\_kWh \right) \end{aligned} \tag{5.4}$$

where $\mathrm{VM}x_{Est\_Cost\_PMi}$ is the estimated total cost of the VM*x* before and during migration on the source PM*i*. The $\mathrm{VM}x_{ReqvCPUs\_PMi}$ is the number of requested vCPUs for the VM and $VMx_{Pred\_U\_PMi}$ is the predicted CPU utilisation for the VM times the cost for requested vCPUs for a period of time.

$\mathrm{VM}x_{Pred\_RAM\_U\_PMi}$ is the predicted memory usage times the cost for that resource for a period of time. We consider the similar notation for the disk and network resources on PM*i*. $\mathrm{VM}x_{Pred\_E\_PMi}$ is the predicted energy consumption of VM*x* times the energy cost as considered by the energy providers.

Similarly, the total cost of the VM*x* during and after migration on the destination PM*j*, $\mathrm{VM}x_{Est\_Cost\_PMj}$, is estimated based on Equation (5.4), but substituting PM*i* with PM*j* and T_{Run_Sou} with T_{Run_Des} for each resource such as CPU, RAM, disk, network and energy.

Thus, the estimated total cost for VM*x*, $\mathrm{VM}x_{Total\_Est\_Cost}$, before and after the migration can be given by:

$$\mathrm{VM}x_{Total\_Est\_Cost} = \mathrm{VM}x_{Est\_Cost\_PMi} + \mathrm{VM}x_{Est\_Cost\_PMj} \quad (5.5)$$

Step 8: Finally, this step compares the estimated total cost of VM*x* before live migration with the estimated total cost of the same VM*x* after the migration takes place, in order to check the ability to recover the costs incurred by live migration, as described in Algorithm 5.3.

Algorithm 5.3: Migration Cost Recovery

Initialise: VM*x* Cost Before Migration = $\mathrm{VM}x_{Est\_Cost\_PMi}$ (as explained in Section 5.2.1 Step 7);
VM*x* Cost After Migration = $\mathrm{VM}x_{Est\_Cost\_PMj}$ (as explained in Section 5.2.1 Step 7).
Input: VMs list.
Output: Boolean Cost Recovery list.

```
for each (VMx in VMs list) do
    if (VMx Cost After Migration ≤ VMx Cost Before Migration) then
        Cost Recovery list = true; // The cost of migration is recovered.
            else
        Cost Recovery list = false; // The cost of migration is not recovered.
    end if
end for
return Cost Recovery list.
```

5.2.2 VMs Auto-Scaling Prediction Models

VMs auto-scaling is an important technique to provide additional capacity to the VMs on-the-fly. Generally, there are two types of VMs auto-scaling [18], [21], [14]: 1) vertical scaling (scale-up): request for more resources (e.g., vCPUs and memory) inside the VMs, and 2) horizontal scaling (scale-out): request for creating additional VMs. However, the latter mechanism may take a few minutes

to initiate [17], [18], [133], [102], which is unacceptable for VMs that need to rapidly scale-out during the computation [19], [20]. Besides, there are additional costs [14] in terms of scaling time (booting/rebooting), license fees for the new VMs (horizontal scaling) and energy overhead that need further consideration [21]. Hence, understanding the impact of VMs auto-scaling is essential for the design of an effective resource provision technique [21].

To enable VMs auto-scaling on-the-fly without any performance loss or delay, some form of prediction mechanism is needed to prepare the VMs in advance [14]. Thus, the proactive framework can help to avoid service performance degradation by taking preventive actions (e.g., VMs auto-scaling) at an early stage. The impact of such a framework will rely on potential actuators/decisions to detect the overloaded VMs in order to decide when to scale, how to scale the VMs and where to scale. Additionally, the proactive framework can assist service providers to estimate the future cost of Cloud services (e.g., VMs auto-scaling) in order to offer suitable services that meet their customers' requirements.

The proposed framework implemented inside the cost modeller (introduced in Section 3.2) has the ability to take advance decisions regarding auto-scaling cost, while considering other quality requirements such as energy consumption and performance of the application [91]. The auto-scaling resource provisioning technique can be described using the MAPE [165] control loop to provision resources when needed, as depicted in Figure 5-9.

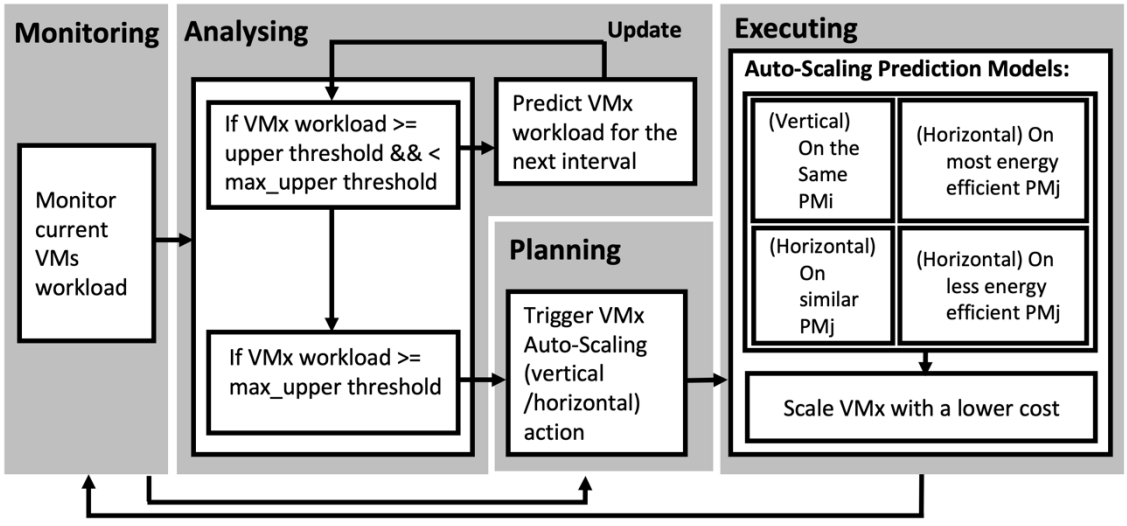


Figure 5-9: Performance and Energy-based Cost Prediction Framework (Auto-Scaling).

The proposed framework is aimed towards predicting workload and power consumption as well as estimating the total cost of the VMs incurred by the auto-scaling decision. To achieve this aim, several steps are required in order to first predict the PMs/VMs workload and power consumption, then estimate the total cost of the scaled VMs as explained next.

Step 1: The VMs CPU utilisation and RAM usage (upper and max_upper) thresholds (e.g., 85% and 95%) are set and the VM*x* workload is monitored periodically to determine whether extra resources are needed. If the VM*x* workload is in the range of [upper and max_upper threshold], then predict the VM*x* workload for the next time interval (e.g., every 5 minutes [164]) using the ARIMA model based on historical workload patterns (as introduced in Chapter 4). This prediction supports the avoidance of the unnecessary scaling caused by the small peaks in the workload (false alarm). If the predicted VM*x* workload for the next interval equals or exceeds the max_upper threshold, VM auto-scaling is performed as described in Algorithm 5.4. The list of the algorithms parameters and their notations is shown in Table 5-1, Section 5.2.1.

Step 2: Algorithm 5.4 is used to identify the overloaded VM*x* to be scaled and potentially the most energy efficient PM*j* to host it, if there is no capacity to perform a vertical scaling in the first place. The VMs are ranked in decreasing order of their workload, whereas the PMs are ranked in decreasing order according to their energy efficiency. The estimation of the energy efficiency for both hosts (source PM*i* and destination PM*j*) can be computed as: PM power = $\frac{\text{PM}i\ (\text{power of the source})}{\text{PM}j\ (\text{power of the destination})}$, as introduced in Section 5.2.1, Step 3. It is also ensured that the destination PMs would have sufficient resources to handle the scaled VM*x* workload in order to prevent service performance degradation (e.g., when VM resource utilisation increases beyond the predefined threshold).

Algorithm 5.4: VMs Workload Prediction and Auto-Scaling Decision

Initialise: VM workload = $\left(\frac{\text{U\_CPU\_VM}}{\text{C\_CPU\_VM}}, \frac{\text{U\_RAM\_VM}}{\text{C\_RAM\_VM}}\right)$;

VM upper threshold = 0.85 × (C_CPU_VM, C_RAM_VM);

VM max_upper threshold = 0.95 × (C_CPU_VM, C_RAM_VM);

PM workload = $\left(\frac{\text{U\_CPU\_PM}}{\text{C\_CPU\_PM}}, \frac{\text{U\_RAM\_PM}}{\text{C\_RAM\_PM}}\right)$;

PM upper threshold = 0.85 × (C_CPU_PM, C_RAM_PM);

PM power = $\frac{\text{PM}i\ (\text{power of the source})}{\text{PM}j\ (\text{power of the candidate})}$; // to check the energy efficiency

Predicted VM workload = null;

VM Resource Increments = (I_CPU_VM, I_RAM_VM) = (null, null);
Scaling Decision = null.
Input: VMs list, PMs list. // Assuming all the PMs in running/active status
Output: Scaling Decision.

1: Sort the PMs list in decreasing order of the PM power;
2: Sort the VMs list on PM*i* in a decreasing order of the workload;
3: **for each** (VM*x* in VMs list) **do**
4: **if** (VM*x* workload ≥ VM*x* upper threshold) && (VM*x* workload < VM*x* max_upper threshold) **then**
5: Predicted VM*x* workload ⟵ predict the (VM*x* workload) for the next interval using the ARIMA model.
6: **if** (Predicted VM*x* workload > VM*x* workload) **then**
7: VM Resource Increments = Predicted VM*x* workload − VM*x* workload;
8: **else**
9: **break**
10: **end if**
11: **end if**
12: **if** (Predicted VM*x* workload ≥ VM*x* max_upper threshold) **then**
13: **if** (PM*i* workload + VM Resource Increments) < PM*i* upper threshold) **then**
// The resource availability on the same host is met (Resize VM*x*)
14: Scaling Decision ⟵ perform VM*x* vertical scaling based on (VM Resource Increments);
15: **break**
16: **else** // Lack of resources on the same host
17: **for each** (PM*j* in PMs list) **do**
18: **if** ((PM*j* workload + VM Resource Increments) < PM*j* upper threshold) **then**
19: Scaling Decision ⟵ perform VM*x* horizontal scaling based on (VM Resource Increments);
// Create a New VM on: 1) the most energy efficient host
or 2) on a similar host configuration to source
or 3) on the less energy efficient host
20: **break**
21: **end if**
22: **end for**
23: **end if**
24: **end if**
25: **return** Scaling Decision.
26: **end for**

Furthermore, this algorithm demonstrates the comparison between vertical scaling (scale-up) and horizontal scaling (scale-out) in order to obtain the most cost-effective scaling decision. The task is to scale the overloaded VM*x* and select the candidate PM to host it. To do so, the following conditions are tested in this order and the subsequent action performed: 1) vertical scaling on the same PM*i* (vertical scaling is limited to the capacity of PM*i* [18], [102], [14]); 2) horizontal scaling on the most energy efficient PM*j*; 3) horizontal scaling on PM*j* which has a similar configuration as the source (e.g., on any PM*j* that has

the same configuration as the source PM*i* in terms of the CPU type and the ratio of idle power), or 4) horizontal scaling takes place on a less energy efficient PM*j*, as illustrated in Figure 5-10.

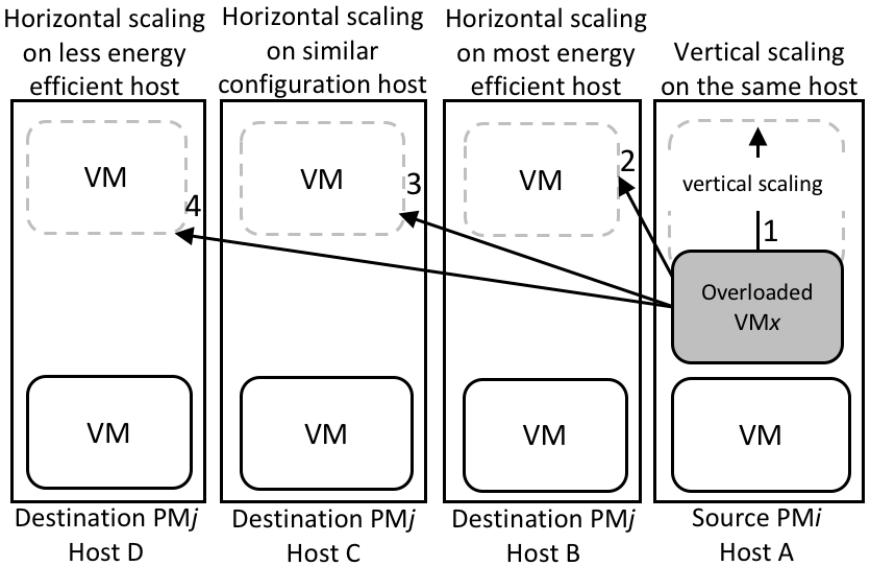


Figure 5-10: The Process of VM Auto-Scaling (Vertical Scaling vs Horizontal Scaling).

Step 3: Algorithm 5.5 is used to select the right size of the VMs to be scaled in a cost-efficient way based on the closest predefined instance sizes set by Cloud providers (e.g., small, medium and large VM). However, this mechanism occasionally leads to resource over-provisioning (e.g., if the requested resources for scaling are less than the predefined instance sizes set by Cloud providers). This may result in resource wasted (needless capacity is created) and the customers might pay more without any benefit [138], [14], [82], which is not the aim of VMs auto-scaling. Moreover, wasted resources may lead to an increase in the cost of energy due to their under-utilisation and a decrease in the Cloud provider's revenue due to the reduction of the number of resource requests that can be accepted. Therefore, a self-configuration approach to resize/create VMs based on the right size of the requested resources is proposed. The self-configuration approach aims to allocate the proper amount of resources to the VMs and avoid the over-provisioning of resources. Thus, this mechanism will help Cloud providers to maximise their resource utilisation beside their profits and the customers will pay for what they actually use, as described in Algorithm 5.5.

Algorithm 5.5: Self-configuration - Resizing/Creating VMs

Initialise: Scaling VM = null.
Input: Scaling Decision; // From Algorithm 5.4 (Vertical or Horizontal Scaling)
VMs size list; // List of VMs sizes set by Cloud providers
VM size. // Based on the predefined VM-sizes list (e.g., small, medium and large)
VM Resource Increments = (I_CPU_VM, I_RAM_VM) // From Algorithm 5.4
Output: Scaling VM.

```
1: Sort the VMs size list in increasing order of the VM sizes;
2: for each (VM size i in VMs size list) do
3:    if (VM Resource Increments = VM size i) then // To ensure that the predefined VM capacity
                                                   is matched with the actual load
4:       Scaling VM = VM size i; // Resize or Create using a predefined VM size based on the Scaling Decision
5:       else
6:          if (VM Resource Increments < VM size i) then
7:             Scaling VM = VM Resource Increments; // Resize or Create using a Self-configuration
                                                    VM size based on the Scaling Decision
8:             break
9:          end if
10:   end if
11: end for
12: return Scaling VM.
```

After identifying the right size of the VM*x* to be scaled and the destination PM*j* to host it, the ARIMA model is used to predict VM*x* workload including (vCPU, memory, disk and network) and identify the best fit model (as introduced in Section 4.2.1).

Once the scaled VM*x* workload is predicted using the ARIMA model based on historical data, the next step is to predict the PMs (source and destination) workload and PMs/VM*x* power consumption using regression models.

Step 4: The prediction of the PMs workload represented as (PMs CPU utilisation), requires measuring the relationship between the number of vCPUs and the PM CPU utilisation for the PMs, as presented in Figures 5-3, 5-4 and 5-5 in Section 5.2.1. The linear regression model of Equation (4.1) presented in Section 4.2.2 is used to predict the PMs CPU utilisation, as mentioned earlier.

Step 5: The PMs power consumption is predicted based on the relationship between the predicted PM workload (PM CPU utilisation) with PM power consumption on the PMs. Using regression analysis, the relation is best described as a linear regression for this particular PM*i*, as presented in Figure 5-

6 in Section 5.2.1. Therefore, in order to predict the PM power consumption, the linear regression model of Equation (4.2) presented in Section 4.2.3, is used.

Not all existing PMs essentially follow a linear power model in relation to their CPU utilisation, as presented in Figures 5-7 and 5-8 in Section 5.2.1. In such a scenario, other regression models can be used to describe the relation between the power consumption and CPU utilisation of the targeted PMs. Therefore, in order to predict the PM power consumption that does not follow the linear model, the polynomial model in Equation (4.3) presented in Section 4.2.3, is used.

Step 6: The proposed Equation (4.4) in Section 4.2.4, is used to predict the VM*x* power consumption on the PMs, then the conversion of the power consumption to energy is required using the Equation (4.5).

Step 7: This step estimates the total cost for the scaled VM*x* based on the predicted VM*x* resource usage in *Step 3* and the predicted VM*x* energy consumption in *Step 6* within this framework.

The total time required for auto-scaling VM*x* can be given by:

$$T_{Scaling\_VMx} = (T_{End\_Scaling} - T_{Start\_Scaling}) \quad (5.6)$$

$$T_{Existing\_VMx} = \left(T_{End\_Run} - T_{Start\_Run}\right) - (T_{Scaling\_VMx}) \quad (5.7)$$

where $T_{Scaling\_VMx}$ is the time required for scaling VM*x* measured by seconds. $T_{Start\_Scaling}$ is the time when the scaling is started and $T_{End\_Scaling}$ is the time when the scaling is ended. $T_{Existing\_VMx}$ is the running time of the existing VM*x* before scaling starts. T_{Start_Run} is the start time of the running task and T_{End_Run} is the end time of the running task.

The total cost for VM*x* would require an estimation of the cost of the VM before and after the scaling process. Hence, to estimate the total cost for VM*x* before scaling, Equation (4.6) presented in Section 4.2.5 is used, but with different notations, as shown in Equation (5.8):

$$\begin{aligned}\mathrm{VM}x_{Est\_Cost\_PMi} = &\left(\left(\mathrm{VM}x_{ReqvCPUs\_PMi} \times \frac{VMx_{Pred\_U\_PMi}}{100}\right)\right. \\ &\left.\times \left(Cost\_vCPU \times T_{Existing\_VMx}\right)\right) \\ &+ \left(\mathrm{VM}x_{Pred\_R\_U\_PMi} \times \left(Cost\_GB \times T_{Existing\_VMx}\right)\right) \\ &+ \left(\mathrm{VM}x_{Pred\_D\_U\_PMi} \times \left(Cost\_GB \times T_{Existing\_VMx}\right)\right) \\ &+ \left(\mathrm{VM}x_{Pred\_N\_U\_PMi} \times \left(Cost\_GB \times T_{Existing\_VMx}\right)\right) \\ &+ \left(\mathrm{VM}x_{Pred\_E\_PMi} \times Cost\_kWh\right)\end{aligned} \quad (5.8)$$

where $\mathrm{VM}x_{Est\_Cost\_PMi}$ is the estimated total cost of VM*x* before scaling on the source PM*i*. The $\mathrm{VM}x_{ReqvCPUs\_PMi}$ is the number of requested vCPUs for the VM and $VMx_{Pred\_U\_PMi}$ is the predicted CPU utilisation for the VM times the cost for requested vCPUs for a period of time. $\mathrm{VM}x_{Pred\_R\_U\_PMi}$ is the predicted resource usage of RAM times the cost for that resource for a period of time before scaling $T_{Existing\_VMx}$. We consider the similar notation for the CPU, disk and network resources on PM*i*. $\mathrm{VM}x_{Pred\_E\_PMi}$ is the predicted energy consumption of VM*x* times the energy cost as considered by the energy providers.

Similarly, the cost of VM*x* after scaling takes place on the destination PM*j* is estimated using Equation (5.8), but substituting PM*i* with PM*j* and $T_{Existing\_VMx}$ with $T_{Scaling\_VMx}$ for each resource such as CPU, RAM, disk, network and energy. Besides, additional license fees α for the new VM is applied when horizontal scaling takes place, and is considered as constant (£0.1/hr).

Thus, the estimated total cost for VM*x* before and after scaling can be given by:

$$\mathrm{VM}x_{Total\_Est\_Cost} = \mathrm{VM}x_{Est\_Cost\_PMi} + \mathrm{VM}x_{Est\_Cost\_PMj} \quad (5.9)$$

5.3 Implementation

The performance and energy-based cost prediction framework is introduced in this research to estimate the total cost of migrated and scaled VMs during service operation. Thus, in order to evaluate this framework, a number of direct experiments have been conducted on the Cloud testbed (see Section 5.3.1) to

synthetically generate historical workload data. The prediction process starts by firstly predicting the PMs/VMs workload using the (***auto.arima***) function in R package [159] to automatically select the best fit model of ARIMA based on AIC or BIC value. Once the PMs/VMs workload is predicted, the process is then going through the steps of the introduced framework to consider the correlation between the physical and virtual resources in order to predict power consumption of the VMs running on multiple PMs. Finally, the total cost of the migrated and scaled VMs is estimated based on their predicted workload and power consumption.

5.3.1 Characterisation of Physical Machines

Four different PMs on the Cloud testbed have been considered. The first three PMs, Host A, C and D, have four core X3430 Intel Xeon CPU, and the last PM, Host B, has an eight-core E3-1230 V2 Intel Xeon CPU. Host A is considered as the source host and Host B, C and D are considered as the destination's hosts. Host B is the most energy efficient host, Host C is the similar host configuration to the source (Host A), and Host D is the less energy efficient host. Also, each PM has a Watt meter [143] attached to directly measure the power consumption. Heterogeneous VMs are created and their monitoring is performed through Zabbix [150], which is also used for resource usage monitoring.

5.4 Experiments and Evaluation

5.4.1 Design of Experiments

The overall aim of the experiments is to demonstrate that the performance and energy-based cost prediction framework is capable of predicting the workload and power consumption as well as estimate the total cost of migrated and scaled VMs when being run on different PMs.

Three direct experiments have been conducted for each live migration and auto-scaling using three types of VMs with the objective to 1) reduce energy-related costs while maintaining performance requirements; 2) estimate the total cost for a number of VMs before and after live migration, in order to check the ability to recover the costs incurred by live migration, and 3) identify the most

suitable cost-effective scaling strategy and estimate the total cost of the scaled VMs accordingly.

To design the experiments, historical data has been generated to represent real workload patterns of Cloud applications (discussed in Section 4.2.1), by using *Stress-ng* tool [73] (see Section 4.4.1) in order to stress all the resources including (CPU, memory, disk and network) on different types of VMs. The generated workload for each VM type has a time interval of four slots (30 minutes each). The first three intervals (slots) are used as the historical data set for prediction, and the last interval (slot) is used as the testing data set to evaluate the predicted results. A similar approach is used in [160] and followed in this thesis.

5.4.2 Evaluation

5.4.2.1 VMs Workload Prediction

This section presents the quantitative evaluation of the performance and energy-based cost prediction framework in terms of VMs live migration and auto-scaling in order to estimate the total cost of VMs during service operation.

Figures 5-11, 5-12, and 5-13 show the predicted workload results for three types of VMs, small, medium and large, running on a multiple PMs based on historical periodic workload pattern. They depict the results of the migrated and scaled VMs predicted versus the actual workload, including CPU, RAM, disk, and network usage for the VMs. Despite the periodic utilisation peaks, the predicted VMs CPU, RAM and network 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. The predicted VMs disk workload is also matching the actual workload, but with less accuracy as compared to the CPU, RAM and network prediction results. This can be justified because of the high variations in the generated historical periodic workload pattern of the disk not closely matching in each interval. Besides the predicted VMs' workload mean values, the results also show the high and low 95% and 80% confidence intervals for the predicted workload of each VM based on the ARIMA model.

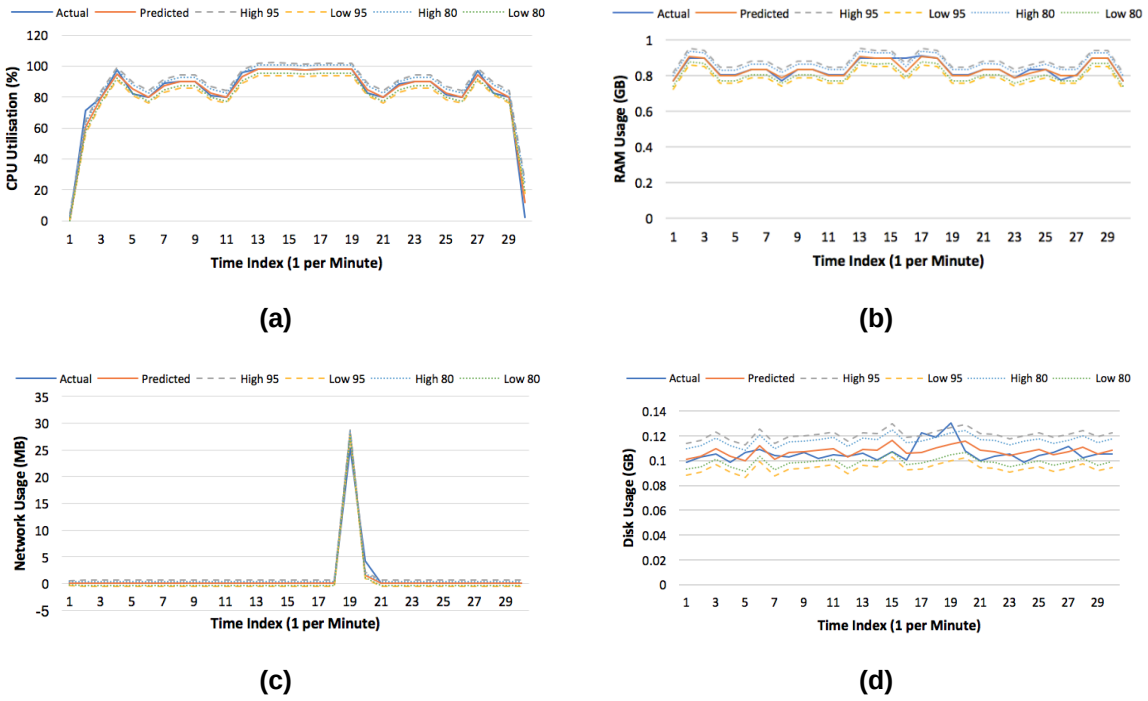


Figure 5-11: The Workload Prediction Results for Small VM.

Table 5-2: Prediction Accuracy for Small VM.

Parameters	ME	RMSE	MAE	MPE	MAPE
CPU Utilisation	0.00486	1.7101	0.5652	-3.4611	4.978
RAM Usage	0.00167	0.0189	0.0055	0.1618	0.6585
Disk Usage	-0.0052	0.1869	0.0461	3.459	6.940
Network Usage	0.00072	0.0051	0.0030	0.64200	2.8612

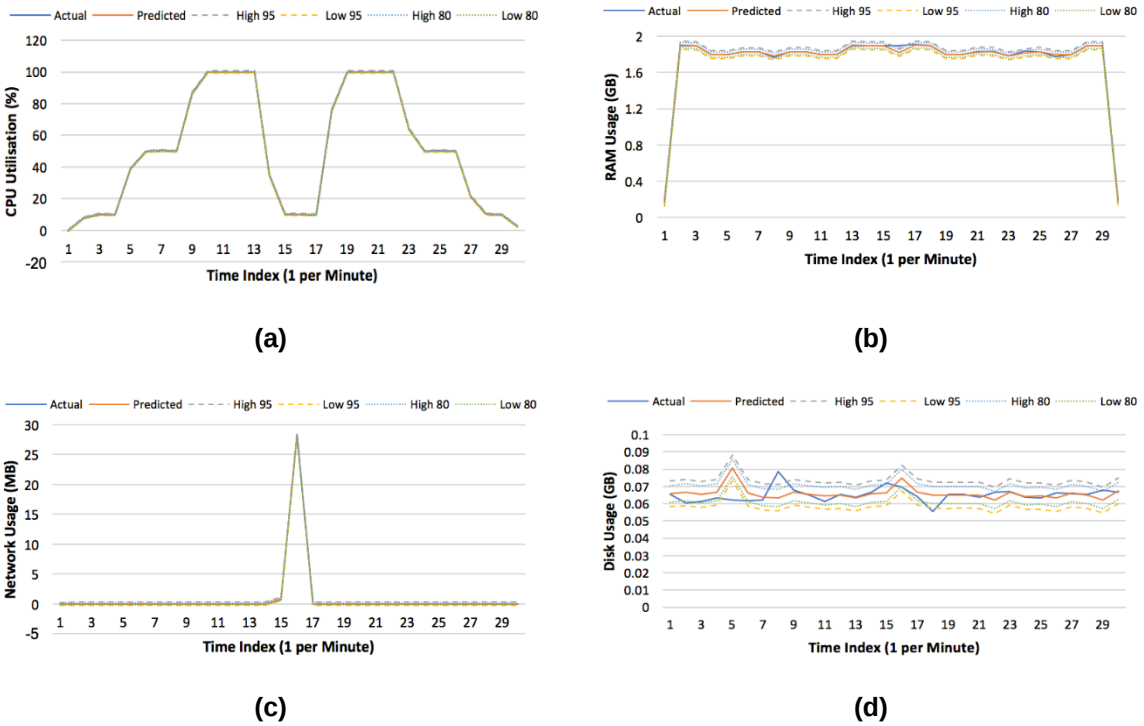


Figure 5-12: The Workload Prediction Results for Medium VM.

Table 5-3: Prediction Accuracy for Medium VM.

Parameters	ME	RMSE	MAE	MPE	MAPE
CPU Utilisation	0.019355	0.2451	0.12275	-3.1443	3.576033
RAM Usage	0.001976	0.0189	0.00588	0.11509	0.333648
Disk Usage	0.000197	0.0940	0.01848	-8.96	9.5482
Network Usage	-0.00005	0.0030	0.00181	-0.2380	2.716369

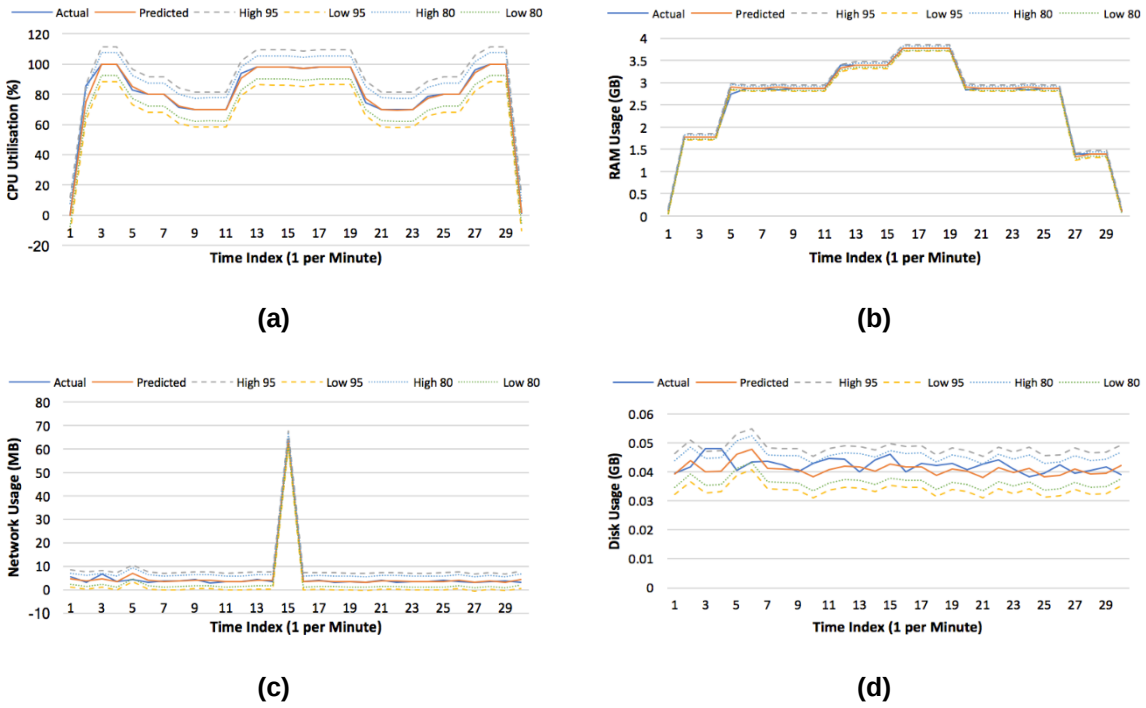


Figure 5-13: The Workload Prediction Results for Large VM.

Table 5-4: Prediction Accuracy for Large VM.

Parameters	ME	RMSE	MAE	MPE	MAPE
CPU Utilisation	0.437240	4.8481	1.39113	0.86261	2.095702
RAM Usage	-0.00097	0.0308	0.00791	-0.0621	0.328699
Disk Usage	-0.08418	1.4943	0.47049	-3.3323	11.57954
Network Usage	-0.00001	0.0028	0.00156	-0.3278	3.637562

In terms of prediction accuracy, a number of metrics have been used to evaluate the results of the predicted workload for three types of VMs, as these metrics have been defined earlier in Section 4.4.2. The accuracy of the predicted VMs workload (CPU, RAM, disk, network) based on periodic workload is evaluated using these accuracy metrics, as summarised in Tables 5-2, 5-3 and 5-4, respectively.

5.4.2.1.1 VMs' Live Migration Workload Prediction

In Algorithm 5.1 of Section 5.2.1, when PM*i* is overloaded and exceeds the predefined (upper threshold), instead of immediately migrating VMs, the prediction model is used to minimise the number of VM migrations and avoid

unnecessary migrations caused by the small peaks in the workload. However, when PM*i* is overloaded and exceeds the predefined (max_upper threshold), the proposed Algorithm 5.2 is used to migrate the candidate VM*x*, in order to reduce the overloaded PM*i* and allocate the VM*x* on appropriate PM*j*, which has sufficient resources and is potentially more energy efficient. It is also ensured that the destination PM*j* utilisation will not exceed the max_upper threshold for reallocating of the incoming VM*x*. To illustrate the migration process, Figure 5-14 shows the predicted versus the actual PMs workload when the VMs run CPU-intensive workload, during the migration process the resource interference incurred by migration has appeared on both source PM*i* and destination PM*j*. Therefore, this resource interference incurred on both source and destination needs to be taken into account [124] when estimating the total migration cost (see Section 5.4.2.3). Moreover, to achieve the migration without degrading the performance, both PM*i* and PM*j* (CPU and RAM) resources need to be carefully managed [124]. Since the PM*i* max_upper threshold is predefined and PM*j* have available resources to accept the candidate VMx, the performance during the migration process is therefore not affected.

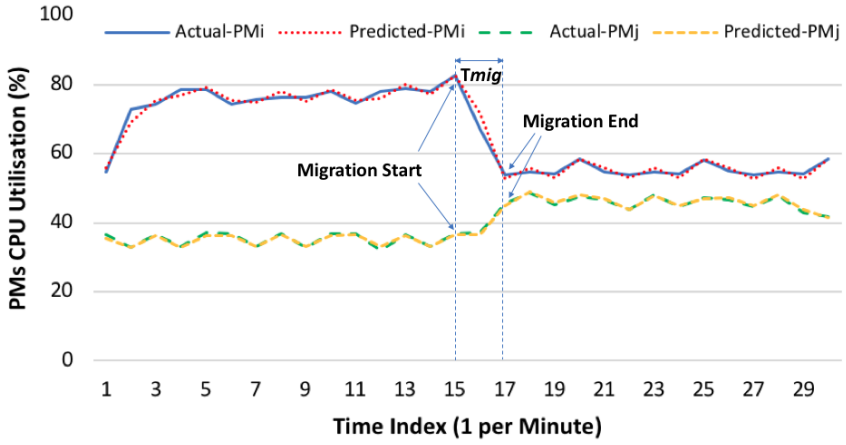


Figure 5-14: The Predicted Workload vs The Actual Workload for both PMs (Source PMi and Destination PMj).

5.4.2.1.2 VMs' Auto-Scaling Workload Prediction

In Algorithm 5.4 of Section 5.2.2, when VM*x* is overloaded and exceeds the predefined (upper threshold), instead of immediately auto-scaling VMs, the prediction model is used to minimise the number of VMs scaling and avoid unnecessary scales caused by the small peaks in the workload. However, when

VM*x* is overloaded and exceeds the predefined (max_upper threshold), the overloaded VM*x* will be scaled in order to prevent service performance degradation and allocated to an appropriate PM*j*, which has sufficient resources and is potentially most energy efficient. To achieve the auto-scaling without degrading the performance of VM*x*, the destination PM*j* (CPU and RAM) resources need to be carefully managed. Since the PM*i* upper threshold is predefined and PM*j* has available resources to accept the allocated VM*x*, the performance of the auto-scaled VM*x* is not affected. It is also ensured that the destination PM*j* will not exceed the upper threshold for allocating of the incoming VM*x*.

5.4.2.2 VMs Power Consumption Prediction

Besides the VMs workload prediction, the proposed framework can predict the power consumption for a number of VMs when running on source PM*i* and destination PM*j* for both live migration and auto-scaling, as described below.

5.4.2.2.1 VMs' Live Migration Power Consumption Prediction

Figures 5-15, 5-16 and 5-17 depict the results of the predicted versus the actual power consumption for a number of VMs when running on source PM*i* (Host A) and destination PM*j,* noting that the destination PM*j* can be the most energy efficient (Host B) or less energy efficient (Host D) comparing to source PM*i* based on the migration decision. Also, the predicted power consumption attribution for each VM is affected by the variation in the predicted CPU utilisation of all the VMs. In terms of prediction accuracy, a number of metrics have been used to evaluate the predicted power consumption for small, medium and large VMs based on periodic workload pattern as presented in Table 5-5.

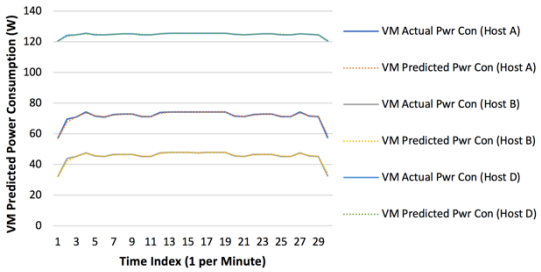


Figure 5-15: Small VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj).

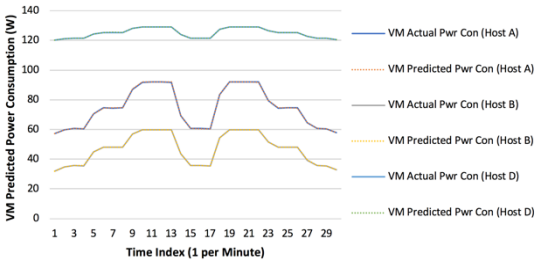


Figure 5-16: Medium VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj).

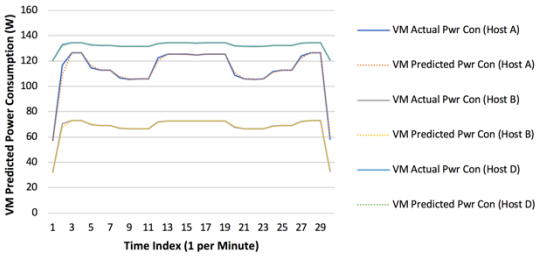


Figure 5-17: Large VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj).

Table 5-5: Prediction Accuracy for The Predicted Power Consumption for all VMs on (Host A, Host B and Host D).

Parameter	VMs	Hosts	ME	RMSE	MAE	MPE	MAPE
VMs Power Consumption	Small VM	Host A	-0.00551665	0.5150904	0.2493285	0.00539324	0.3674461
		Host B	0.005655233	0.4750381	0.2190667	0.04799226	0.5281281
		Host D	0.00246747	0.1537848	0.07028654	0.0023619	0.05689478
	Medium VM	Host A	0.01939327	0.07113483	0.04306951	0.02648363	0.05983904
		Host B	0.01529777	0.05683427	0.03492552	0.03521646	0.08024377
		Host D	0.004925887	0.01823638	0.01120869	0.003956332	0.00901164
	Large VM	Host A	-0.2564522	1.533448	0.5685501	-0.2213621	0.5101096
		Host B	-0.07265782	0.5223516	0.193443	-0.0954475	0.2912161
		Host D	0.00000132	0.0000031	0.00000278	0.00000099	0.0000021

5.4.2.2.2 VMs' Auto-Scaling Power Consumption Prediction

Figures 5-18 to 5-23 show the results of the predicted versus the actual power consumption for the VMs running on a number of PMs using different scaling strategies based on the predefined instance size and the self-configuration instance size (as discussed in Algorithm 5.5 of Section 5.2.2). According to Algorithm 5.5, the vertical scaling is performed on the same PM (Host A) and the horizontal scaling is performed on a number of hosts, (Host B) is the most energy efficient PM, (Host C) is a similar host configuration to the source (Host A), and

(Host D) is the less energy efficient PM. Note that, the vertical scaling was not performed with the large VM, since the large VM has four CPU cores and the capacity of the hosted PM (Host A) has the same number of CPU cores as the VM. Thus, there is no available capacity to perform vertical scaling on the same host. Therefore, only horizontal scaling can be performed with this specified VM.

By observing the figures, the self-configuration auto-scaling mechanism proposed in (Algorithm 5.5) outperforms the predefined one, since the predicted power consumption is lower, thus the total cost of VMs will be lower as well. Also, it should be mentioned that the predicted power consumption attribution for each VM is affected by the variation in the predicted PM CPU utilisation of all the VMs. In terms of prediction accuracy, a number of metrics have been used to evaluate the predicted power consumption for small, medium and large VMs based on periodic workload pattern as presented in Table 5-6.

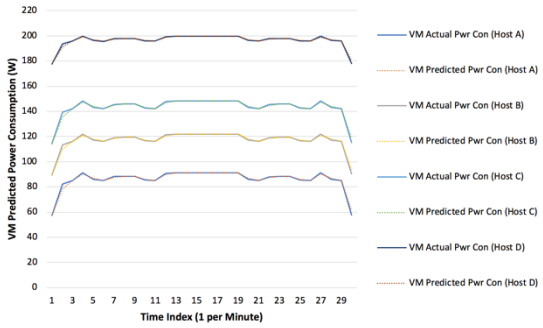


Figure 5-18: Small VM Predicted vs Actual Power Consumption using a Predefined VM Size - Scaling on a Number of PMs.

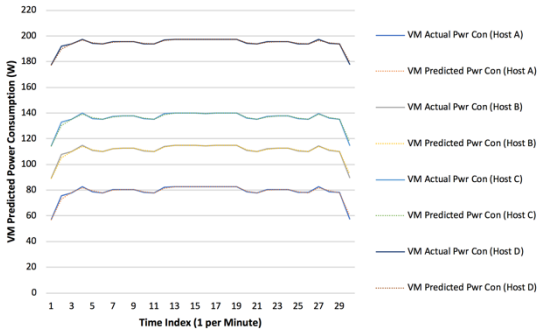


Figure 5-19: Small VM Predicted vs Actual Power Consumption using Self-Configuration VM Size - Scaling on a Number of PMs.

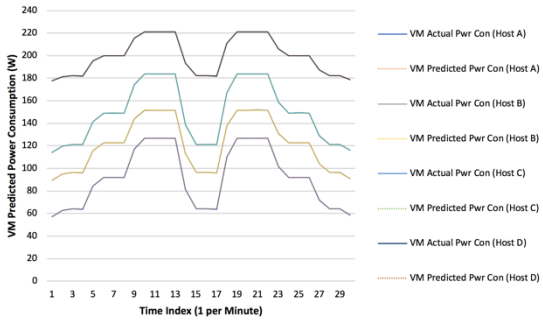


Figure 5-20: Medium VM Predicted vs Actual Power Consumption using a Predefined VM Size - Scaling on a Number of PMs.

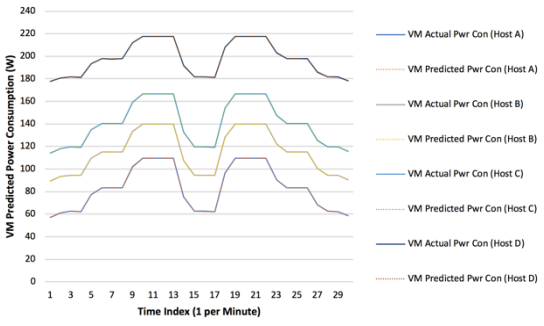


Figure 5-21: Medium VM Predicted vs Actual Power Consumption using Self-Configuration VM Size - Scaling on a Number of PMs.

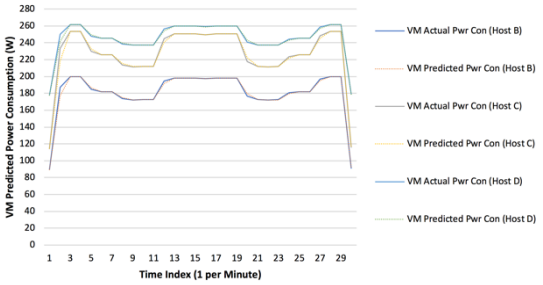


Figure 5-22: Large VM Predicted vs Actual Power Consumption using a Predefined VM Size - Scaling on a Number of PMs.

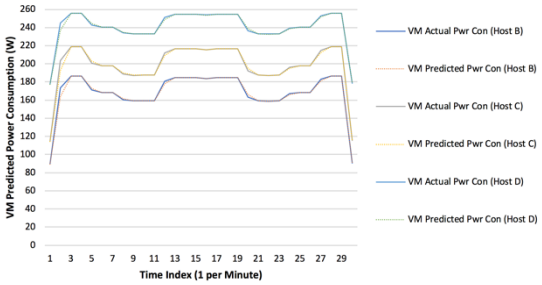


Figure 5-23: Large VM Predicted vs Actual Power Consumption using Self-Configuration VM Size - Scaling on a Number of PMs.

Table 5-6: Prediction Accuracy The Predicted Power Consumption for all VMs on (Host A, Host B, Host C and Host D).

Parameter	VMs	Hosts	ME	RMSE	MAE	MPE	MAPE
VMs Power Consumption	Small VM	Host A	-0.01103331	1.030181	0.498657	0.03021525	0.6409635
		Host B	0.000138589	0.9890217	0.4683952	0.02041722	0.4281543
		Host C	-0.01103331	1.030181	0.498657	0.00539323	0.3674461
		Host D	-0.00304837	0.6681303	0.3196148	0.0011727	0.1668069
	Medium VM	Host A	0.03878653	0.1422697	0.08613903	0.04478487	0.1014052
		Host B	0.03469103	0.1274472	0.07799503	0.0297295	0.06752187
		Host C	0.03878655	0.1422697	0.08613901	0.02648364	0.05983903
		Host D	0.02431962	0.08910762	0.05427882	0.01224154	0.02755249
	Large VM	Host B	-0.32911	2.054355	0.7619931	-0.17381	0.4269158
		Host C	-0.5129043	3.066896	1.1371	-0.2213621	0.5101096
		Host D	-0.2934642	1.762708	0.6549309	-0.1180191	0.2683187

5.4.2.3 VMs Total Cost Estimation

This framework is also capable of estimating the live migration and auto-scaling total cost for a number of VMs when running on different PMs.

5.4.2.3.1 VMs' Live Migration Cost Estimation

Figures 5-24 and 5-25 show the results of the estimated total cost before and after live migration for a number of VMs along with their migration cost recovery based on the proposed Algorithm 5.3 in Section 5.2.1. In Figure 5-24, the estimated migration cost recovery can be achieved for all three types of VMs when the migration is performed to the most energy efficient host. Conversely,

when the migration is performed to the less energy efficient host as shown in Figure 5-25, only a large VM can recover that migration cost.

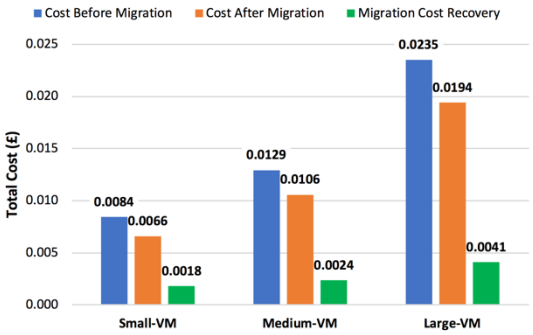


Figure 5-24: Estimated Total Cost Before vs After Migration with Migration Cost Recovery on (most energy efficient PM), Host B.

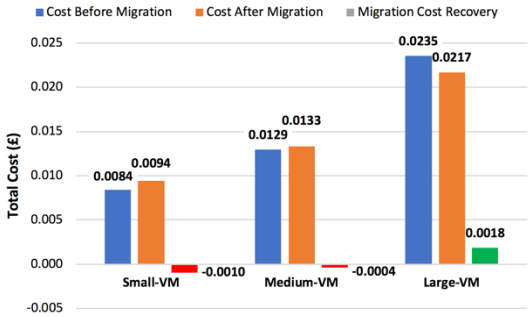


Figure 5-25: Estimated Total Cost Before vs After Migration with Migration Cost Recovery on (less energy efficient PM), Host D.

In addition, Figures 5-26 and 5-27 show the percentage results of the estimated migration cost recovery for all three types of VMs when being migrated to the most energy efficient host: 21.53% for the small VM, 18.20% for the medium VM and 17.44% for the large one. However, when the VMs are migrated to the less energy efficient host, only the large VM can recover that migration cost with 7.81%, for small and medium VMs the migration cost cannot be recovered (-11.35% and -2.95%), respectively.

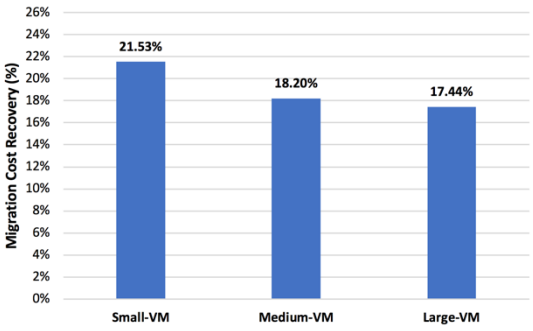


Figure 5-26: The Potential Migration Cost Recovery on (most energy efficient PM), Host B.

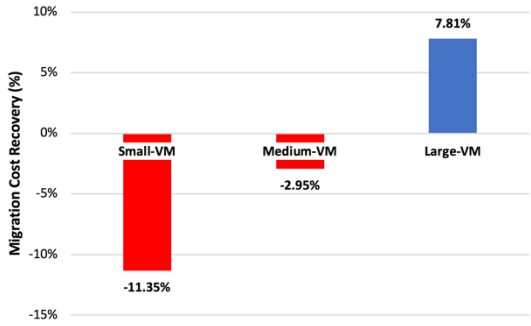


Figure 5-27: The Potential Migration Cost Recovery on (less energy efficient PM), Host D.

5.4.2.3.2 VMs' Auto-Scaling Cost Estimation

Figures 5-28, 5-29 and 5-30 show the results of the estimated auto-scaling total cost for three types of VMs running on a number of PMs using different scaling

strategies, along with the self-configuration technique (proposed in Algorithm 5.5 of Section 5.2.2). This helps select the most suitable cost-effective scaling strategy.

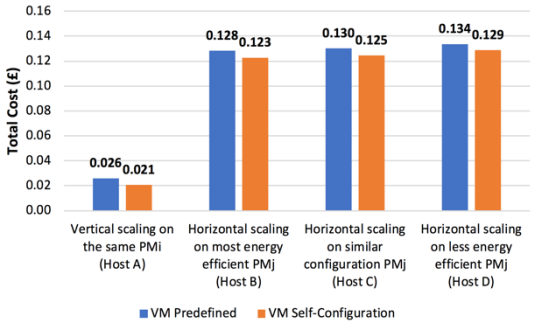


Figure 5-28: Estimated Small VM Auto-Scaling Total Cost (Predefined VM Size Scaling vs Self-Configuration VM Size Scaling).

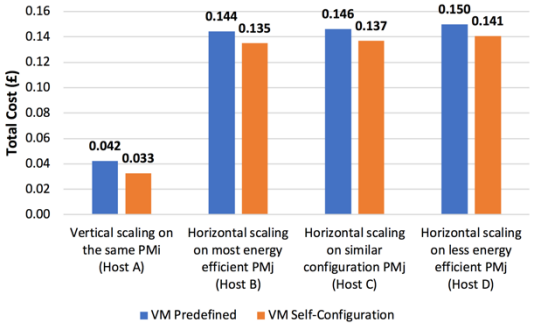


Figure 5-29: Estimated Medium VM Auto-Scaling Total Cost (Predefined VM Size Scaling vs Self-Configuration VM Size Scaling).

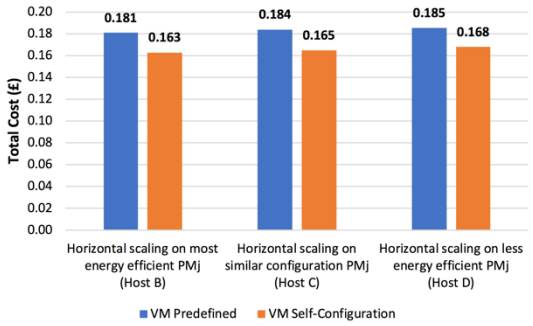


Figure 5-30: Estimated Large VM Auto-Scaling Total Cost (Predefined VM Size Scaling vs Self-Configuration VM Size Scaling).

As shown in Figures 5-28, 5-29 and 5-30, choosing between vertical and horizontal scaling can have a significant impact on the cost of the scaled VMs (e.g., vertical scaling can be more cost-effective than horizontal scaling when the VM scaled on a similar host configuration). This can be justified because of the additional cost in terms of software license for the new VM when horizontal scaling is performed. Also, horizontal scaling using most energy efficient PM can be more cost-effective than horizontal scaling when using less energy efficient PM. As mentioned earlier, the vertical scaling was not performed with the large VM (see Figure 5-30), since it has the same number of CPU cores as the hosted PM (Host A). This means that the host is fully utilised via the VM.

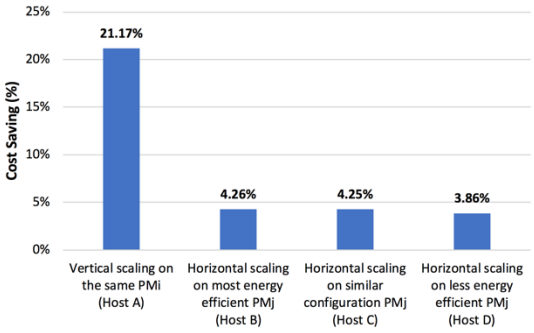


Figure 5-31: Cost Saving by Self-Configuration Scaling (Small VM).

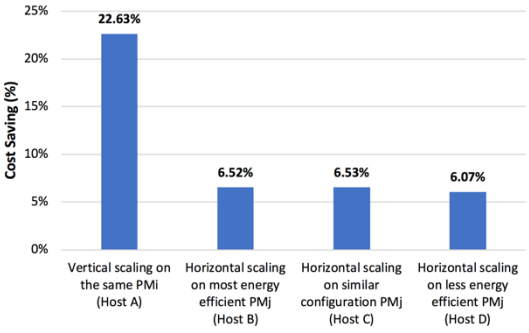


Figure 5-32: Cost Saving by Self-Configuration Scaling (Medium VM).

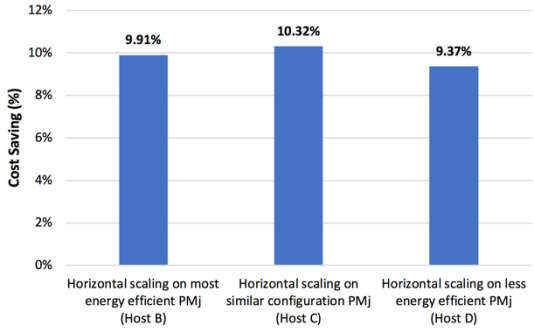


Figure 5-33: Cost Saving by Self-Configuration Scaling (Large VM).

In addition, Figures 5-31, 5-32 and 5-33 show the results of the estimated self-configuration cost that can incur less VMs scaling cost compared to predefined instance size choices. The cost comparison shows that choosing a self-configuration VMs size can achieve about 21.9% cost-saving compared to the predefined VMs size on the same host (PM*i*) when vertical scaling is performed. In case of horizontal scaling on (PM*j*), around 6.89% cost-saving can be gained on a most energy efficient host, approximately 7.03% on a similar host configuration and about 6.43% on a less energy efficient host.

5.5 Summary

This chapter has presented and evaluated a new performance and energy-based cost prediction framework that dynamically supports VMs live migration and auto-scaling to demonstrate the trade-off between cost, power consumption, and performance during service operation. This framework estimates live migration and auto-scaling total cost for heterogeneous VMs by considering their resource

usage and power consumption, while at the same time maintaining the expected level of application performance. The results show that the proposed framework can predict the resource usage, power consumption and estimate the total cost for the migrated and scaled VMs based on historical workload patterns, when being run on heterogeneous PMs.

Chapter 6. A Hybrid Approach for Performance and Energy-based Cost Prediction

6.1 Overview

In this chapter, a new hybrid approach for performance and energy-based cost prediction that aims to integrate auto-scaling with live migration in order to estimate the total cost of VMs by considering resource usage and power consumption is presented in Section 6.2. This approach works by detecting the underloaded and overloaded PMs in order to perform the most cost-effective decision(s) to handle the service performance variation. A number of experiments along with their results are presented in Sections 6.3 and 6.4 to evaluate the capability of this hybrid approach to predict the workload, power consumption and estimate the total cost of VMs scaling and migration when being run on different PMs at service operation.

6.2 Integration of VMs Auto-Scaling with Live Migration: A Hybrid Approach

In a Cloud environment, resource provisioning and VMs consolidation are used to address workload fluctuations issues. Current solutions such as resource provisioning attempt to provide additional resource capacity to the VMs as needed in order to meet QoS requirements. For instance, when one or more VMs are detected as overloaded (e.g., the workload exceeds the predefined upper threshold), the VMs should be scaled up/out to meet the application demands. This can be achieved either by vertical scaling (adds resources to the VMs on the same host) or by horizontal scaling (creates new VMs on appropriate host), all of which were based on application requirements. Another solution such as VM consolidation aims to move the VMs from one host to another in order to reduce the number of active PMs and save power. For instance, when a host is detected as underloaded (e.g., the workload less than the predefined lower threshold), it is a candidate for being switched off or to enter power saving mode. This can be accomplished by re-allocating VMs through live migration to an

appropriate host. However, these techniques have their own set of limitations in terms of the additional costs related to scaling/migration time and energy consumption, as discussed earlier in Chapter 5.

A number of papers in the literature [12]–[14], [20], [21], [100], [101], [113] have investigated the resource provisioning and VM consolidation independently through different objectives such as load balancing, increasing the capacity of VMs resources and reducing the energy-related costs. To minimise the operational costs while achieving performance objectives, Cloud providers can automatically perform an integration of VMs consolidation and resource provisioning to match workload changes and prevent any performance loss. Thus, a proactive framework has the advantage of taking preventive actions on-the-fly (e.g., VMs auto-scaling, migrating and re-allocating) at earlier stages to avoid service performance degradation. The effectiveness of such framework depends on potential actuators/decisions to implement at service operation. This solution would allow Cloud providers to make better use of their infrastructure in terms of maintaining service performance, reducing power consumption and operating cost. In addition, estimating the future cost of Cloud services can help the service providers offer suitable services that meet their customers' requirements.

Therefore, the proposed framework (discussed in Chapter 5) has been extended to support a new hybrid approach for performance and energy-based cost prediction, as depicted in Figure 6-1. This approach is implemented inside the cost modeller (introduced in Section 3.2) and supports decision-making regarding auto-scaling and live migration, considering their costs, while at the same time being aware of the impact on other quality characteristics such as energy consumption and performance of the application.

This approach is aimed towards predicting PMs/VMs workload using the ARIMA model in order to perform the most effective decision(s) (e.g., auto-scaling, live migration or both) to handle the performance variation of the applications. The relationship between the VMs and PMs workload is investigated using regression models in order to predict the VMs power consumption for an efficient allocation/re-allocation of the VMs. Hence, the total cost of the VMs incurred by the most effective decision(s) can be estimated based on the predicted workload and energy consumption for each VM.

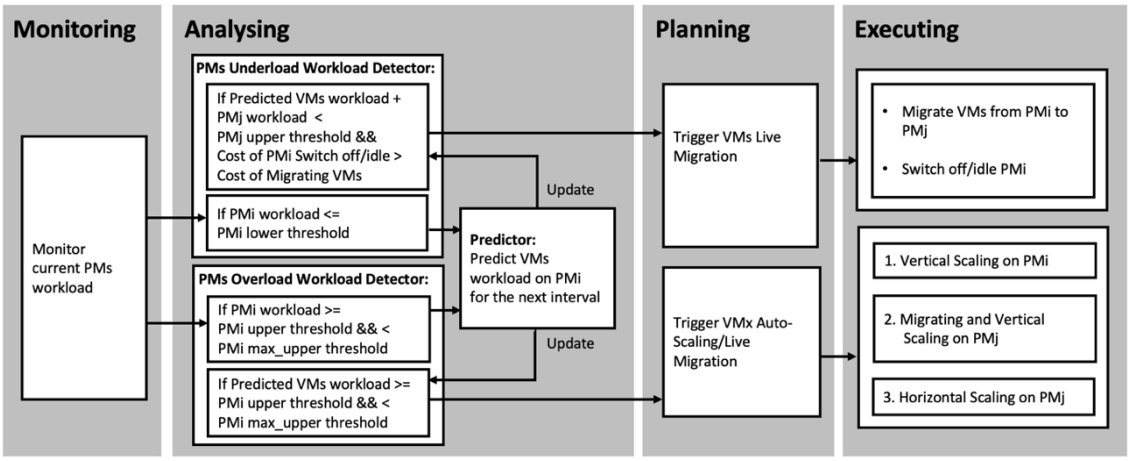


Figure 6-1: A Hybrid Approach for Performance and Energy-based Cost Prediction.

To achieve this aim, several steps are required in order to detect the underloaded and overloaded PMs, predict the PMs/VMs workload and their power consumption, then estimate the total cost of the scaled/migrated VMs as explained below.

Step 1: The PMs CPU utilisation and RAM usage (lower, upper and max_upper) thresholds (e.g., 25%, 85% and 95%, respectively) are set and the PMs workload is monitored periodically. The proposed Algorithm 6.1 is used to detect the underloaded and overloaded PMs. This algorithm combines two sub-algorithms: 1) live migration with VMs re-allocation in order to switch the underloaded host to power saving mode, hence save energy-related costs. Also, this sub-algorithm aims to minimise the overall cost of migration by re-allocation the VMs to the most energy efficient host (if possible), as presented in Algorithm 6.2; and 2) an integration of auto-scaling, live migration and re-allocation in order to prevent the host to be overloaded. This sub-algorithm would help to select the most cost-effective action(s) in order to minimise the overall cost of the VMs incurred by scaling/migration decision(s), as presented in Algorithm 6.3. The list of the algorithms parameters and their notations was shown in Table 5-1 of Chapter 5.

Step 2: In terms of the underloaded PMs, Algorithm 6.1 is used to detect the underloaded PMs and perform appropriate actions such as live migration and re-allocation to save energy cost. Therefore, if the PM*i* workload ($\sum_{i=1}^{n} \mathrm{VM}s\ \mathrm{workload}$) is less than or equals to the lower threshold (e.g., 25%), then predict the VMs workload for the next time interval (e.g., every 5 minutes)

using the ARIMA model based on historical workload patterns (see Step 4). This prediction helps detect the underloaded PM*i* workload in order to migrate the VMs and switch PM*i* to power saving mode. Thus, if the predicted VMs workload for the next interval is still less than or equals to the lower threshold, then VMs live migration and re-allocation are performed using Algorithm 6.2.

Algorithm 6.1: PMs Underload/Overload Workload Detector and Performance Prediction

Initialise: PM workload = $\left(\frac{\text{U\_CPU\_PM}}{\text{C\_CPU\_PM}}, \frac{\text{U\_RAM\_PM}}{\text{C\_RAM\_PM}}\right)$;
PM lower threshold = 0.25 × (C_CPU_PM, C_RAM_PM);
PM upper threshold = 0.85 × (C_CPU_PM, C_RAM_PM);
PM max_upper threshold = 0.95 × (C_CPU_PM, C_RAM_PM);
VM workload = $\left(\frac{\text{U\_CPU\_VM}}{\text{C\_CPU\_VM}}, \frac{\text{U\_RAM\_VM}}{\text{C\_RAM\_VM}}\right)$;
Predicted VM workload = null;
Predicted $\sum_{i=1}^{n}$ VMs workload = null;
Predicted VMs list workload = empty.
Input: PMs list.

1: **for each** (PM*i* in PMs list) **do**
2: **if** (PM*i* workload ≤ PM*i* lower threshold) **then**
3: **for each** (VM*x* in PM*i*) **do**
4: Predicted VM*x* workload ⟵ predict VM*x* workload for the next interval using the ARIMA model.
5: Predicted $\sum_{i=1}^{n}$ VMs workload ⟵ Predicted VM*x* workload ++; // The sum of the predicted VMs workload on PM*i*.
6: **end for**
7: **if** (Predicted $\sum_{i=1}^{n}$ VMs workload ≤ PM*i* lower threshold) **then** // Underloaded PM
8: Perform Algorithm 6.2.
9: **end if**
10: **else**
11: **if** (PM*i* workload ≥ PM*i* upper threshold) && (PM*i* workload < PM*i* max_upper threshold) **then**
12: **for each** (VM*x* in PM*i*) **do**
13: Predicted VM*x* workload ⟵ predict VM*x* workload for the next interval using the ARIMA model.
14: Predicted $\sum_{i=1}^{n}$ VMs workload ⟵ Predicted VM*x* workload ++; // The sum of the predicted VMs workload on PM*i*.
15: Predicted VMs list workload ⟵ Predicted VM*x* workload ++; // The list of the predicted VMs workload on PM*i*.
16: **end for**
17: **if** (Predicted $\sum_{i=1}^{n}$ VMs workload ≥ PM*i* upper threshold) && (Predicted $\sum_{i=1}^{n}$ VMs workload < PM*i* max_upper threshold) **then** // Overloaded PM
18: Perform Algorithm 6.3.
19: **end if**
20: **end if**
21: **end if**
22: **end for**

Algorithm 6.2: Switching PMs to Power Saving Mode

Initialise: PM workload = $\left(\frac{\text{U\_CPU\_PM}}{\text{C\_CPU\_PM}}, \frac{\text{U\_RAM\_PM}}{\text{C\_RAM\_PM}}\right)$;

PM upper threshold = 0.85 × (C_CPU_PM, C_RAM_PM);

PM power = $\frac{\text{PM}i \text{ (power of the source)}}{\text{PM}j \text{ (power of the candidate)}}$; // To check the energy efficiency

Decision = null.

Input: PMs list; // Assuming all the PMs in running/active state

Predicted $\sum_{i=1}^{n}$ VMs workload; // From Algorithm 6.1

Cost of Switching PM*i* to Power Saving Mode;

Cost of Migrating VMs. // To any targeted host PM*j*

Output: Decision.

```
1: Sort the PMs list in decreasing order of the PM power;
2:     for each (PMj in PMs list) do
3:         if ((Predicted ∑_{i=1}^{n} VMs workload + PMj workload) < PMj upper threshold)
               &&
               (Cost of Switching PMi to Power Saving Mode > Cost of Migrating VMs) then
4:             Decision ⟵ perform VMs migration to target host PMj;
5:             Switch PMi to Power Saving Mode.
6:             break
7:         end if
8:     end for
9: return Decision.
```

The proposed Algorithm 6.2 is used to select a matching destination PM*j* to host the migrated VMs, checking whether the cost incurred by VMs live migration is less than the cost of switching the source PM*i* to power saving mode. To do so, the PMs are ranked in decreasing order according to their energy efficiency. This is aimed at migrating the VMs to the most energy efficient host. In this regard, the estimation of the energy efficiency for both source PM*i* and destination PM*j* are considered (as described in Section 5.2.1). Starting with the PM*j* with the lowest idle power (the most energy efficient host), PM*j* is checked whether it has enough resources to meet the migration requirements while at the same time ensuring that the destination host PM*j* will not exceed the upper threshold for allocating of the migrated VMs.

This algorithm ensures: 1) the migrated VMs do not overload the destination PM*j*, 2) the source PM*i* will be switched to power saving mode once the migration takes place in order to save energy cost.

Step 3: In terms of the overloaded PMs, Algorithm 6.1 is used to detect the overloaded PMs and identify the candidate VMs that need to be

scaled/migrated. Therefore, if the PM*i* workload is in the range of [upper and max_upper threshold], then the VMs workload is predicted in PM*i* for the next time interval (e.g., every 5 minutes) using the ARIMA model based on historical workload patterns (see Step 4). This prediction helps to detect in advance the overloaded PM*i* workload and perform preventive actions such as VMs auto-scaling and live migration. Further, this algorithm would help to control the number of scaling and migrations decisions in order to avoid unnecessary scaling/migration caused by the small peaks in the workload (false alarm). Thus, if the predicted VMs workload for the next interval is still in the range of [upper and max_upper threshold], VMs auto-scaling/live migration is performed using Algorithm 6.3.

The proposed Algorithm 6.3 combines the auto-scaling (vertical/horizontal scaling) with live migration in order to obtain the most cost-effective decision(s). The task is to scaling/migrate the overloaded VMs (e.g., resize VMs, migrate existing VMs and resize them, or initiate new VMs), then select appropriate destination PM*j* to host it. To do so, the following conditions are tested in this order and the subsequent actions performed:

1) if PM*i* (the source host) has enough resources to meet the scaling requirements, the vertical scaling is performed on the same PM*i* (hint: vertical scaling is limited to the capacity of PM*i* [18], [102], [14]);
2) in the case if PM*i* does not have enough resources, the PMs are ranked in decreasing order according to their energy efficiency, as described in Section 5.2.1. After sorting the PMs based on their energy efficiency, the *migration and vertical scaling* decision is performed in order to firstly *migrate* the overloaded VMs to appropriate host PM*j* and then *vertically scaling* them. It also checks if PM*j* has enough resources to meet the *migration* and *scaling* requirements while at the same time ensuring that the destination host PM*j* will not exceed the upper threshold for allocating of the migrated VMs along with their scaling requirements (the additional resources); otherwise
3) horizontal scaling takes place on PM*j* in a similar manner as the previous action by placing the new VM to an appropriate destination.

This algorithm ensures: 1) the scaling and migrations VMs do not overload the destination PM*j*, 2) the source PM*i* workload decreases significantly once scaling/migration has taken place, and 3) minimise the overall cost of the VMs incurred by scaling/migration decisions.

Algorithm 6.3: Integrate Auto-Scaling Decisions with Dynamic VMs Allocation

Initialise: PM workload = $\left(\frac{\text{U\_CPU\_PM}}{\text{C\_CPU\_PM}}, \frac{\text{U\_RAM\_PM}}{\text{C\_RAM\_PM}}\right)$;

PM upper threshold = 0.85 × (C_CPU_PM, C_RAM_PM);

PM max_upper threshold = 0.95 × (C_CPU_PM, C_RAM_PM);

PM power = $\frac{\text{PM}i \text{ (power of the source)}}{\text{PM}j \text{ (power of the candidate)}}$; // To check the energy efficiency

Candidate PM = false;

VM workload = $\left(\frac{\text{U\_CPU\_VM}}{\text{C\_CPU\_VM}}, \frac{\text{U\_RAM\_VM}}{\text{C\_RAM\_VM}}\right)$; // From Algorithm 6.1

Candidate VM = false;

Overloaded VM = null.

VM Resource Increments = (I_CPU_VM, I_RAM_VM) = (null, null);

Decision = null.

Input: PMs list; // Assuming all the PMs in running/active state

Predicted VMs list workload; // From Algorithm 6.1

Predicted VM workload; // From Algorithm 6.1

Output: Decision.

```
Sort the Predicted VMs list workload on PMi in a decreasing order;
    for each (VMx in Predicted VMs list workload) do
        if (Predicted VMx workload > VMx workload) then
            Overloaded VMx = Predicted VMx workload;
            VMx Resource Increments = Predicted VMx workload − VMx workload;
            Candidate VM = true;
            break
        end if
    end for
if (Candidate VM = false) then
    break // no candidate VM is found
else
    if ((PMi workload + VMx Resource Increments) < PMi max_upper threshold) then
        // The resource availability on the same host is met (Resize VMx)
        Decision ← perform VMx vertical scaling based on (VMx Resource Increments);
    else // Lack of resources on the same host
        Sort the PMs list in decreasing order of the PM power;
        for each (PMj in PMs list) do
            if ((PMj workload + Overloaded VMx) < PMj upper threshold) then
                Decision ← perform VMx migration to target host PMj; and
                           perform VMx vertical scaling based on (VMx Resource Increments);
                           // Migrate existing VM and resize it
                Candidate PM = true;
                break
            end if
```

```
23:            end for
24:          if (Candidate PM = false) then
25:            for each (PMj in PMs list) do
26:              if ((PMj workload + VMx Resource Increments) < PMj upper threshold) then
27:                Decision ⟵ perform VMx horizontal scaling based on (VMx Resource Increments);
                                // Create a New VM
28:                break
29:              end if
30:            end for
31:          end if
32:      end if
33:  end if
34:  return Decision.
```

Step 4: The ARIMA model is used to predict the VMs workload including (vCPU, memory, disk and network) for the next time interval and identify the best fit model (as introduced in Section 4.2.1). The prediction will help perform the most suitable action(s) and scale the VMs in a cost-efficient way based on the right size of the requested resources using the self-configuration technique (as described in Algorithm 5.5). Once the VMs workload is predicted using the ARIMA model based on historical data, the next step is to predict the PMs (source and destination) workload and PMs/VMs power consumption using regression models.

Step 5: To predict the PMs workload represented as (PMs CPU utilisation), would require measuring the relationship between the number of vCPU and the PM CPU utilisation for the PMs, as presented in Figures 5-3, 5-4 and 5-5 in Section 5.2.1. The linear regression model of Equation (4.1) presented in Section 4.2.2 is used to predict the PMs CPU utilisation.

Step 6: The PMs power consumption is predicted based on the relationship between the predicted PM workload (PM CPU utilisation) with PM power consumption on the PMs. Using a regression analysis, the relation is best described as linear regression for this particular PM*i*, as presented in Figure 5-6 in Section 5.2.1. The linear regression model of Equation (4.2) presented in Section 4.2.3 is used to predict the PMs power consumption.

As discussed earlier in Chapters 4 and 5, not all existing PMs necessarily follow a linear power model in relation to their CPU utilisation, as presented in Figures 5-7 and 5-8 in Section 5.2.1. In this case, other regression models, such

as polynomial, can be used to characterise the relation between the power consumption and CPU utilisation of the targeted PMs, as presented in Equation (4.3), Section 4.2.3.

Step 7: The proposed Equation (4.4) presented earlier in Section 4.2.4 is used to predict the VM*x* power consumption on the PMs, then the conversion of the power consumption to energy is required using the Equation (4.5).

Step 8: Finally, this step estimates the total cost for the VM*x* based on the predicted VM*x* resource usage in *Step 4* and the predicted VM*x* energy consumption in *Step 7*.

The total time, $Time_s$, required for migrating VM*x* can be obtained using Equations (5.1), (5.2), and (5.3) presented in Section 5.2.1. Also, the total time, $Time_s$, required for auto-scaling VM*x* can be obtained using Equations (5.6), and (5.7) presented in Section 5.2.2.

To estimate the total cost for VM*x* based on the performed action(s), Equation (4.6) presented in Section 4.2.5 is used, but with different notations, as shown in Equation (6.1):

$$\begin{aligned} \mathrm{VM}x_{Est\_Cost\_PMi} = & \left(\left(\mathrm{VM}x_{ReqvCPUs\_PMi} \times \frac{VMx_{Pred\_U\_PMi}}{100} \right) \right. \\ & \left. \times (Cost\_vCPU \times Time_s) \right) \\ & + \left(\mathrm{VM}x_{Pred\_RAM\_U\_PMi} \times (Cost\_GB \times Time_s) \right) \\ & + \left(\mathrm{VM}x_{Pred\_Disk\_U\_PMi} \times (Cost\_GB \times Time_s) \right) \\ & + \left(\mathrm{VM}x_{Pred\_Network\_U\_PMi} \times (Cost\_GB \times Time_s) \right) \\ & + \left(\mathrm{VM}x_{Pred\_Energy\_PMi} \times Cost\_kWh \right) \end{aligned} \tag{6.1}$$

where $\mathrm{VM}x_{Est\_Cost\_PMi}$ is the estimated total cost of the VM*x* before and during the action(s) takes place on the source PM*i*. The $\mathrm{VM}x_{ReqvCPUs\_PMi}$ is the number of requested vCPUs for the VM and $\mathrm{VM}x_{Pred\_U\_PMi}$ is the predicted utilisation for the VM times the cost for requested vCPUs for a period of time, (the time, $Time_s$, is based on the performed action, it can be “migration or scaling or both”). $\mathrm{VM}x_{Pred\_RAM\_U\_PMi}$ is the predicted memory usage times the cost for that resource for a period of time. We consider the similar notation for disk and

network resources. $\mathrm{VM}x_{Pred\_Energy\_PMi}$ is the predicted energy consumption of VM*x* times the energy cost as considered by the energy providers.

Similarly, the cost of the VM*x* during and after the action(s) takes place on the destination PM*j* will be estimated using Equation (6.1), but substituting PM*i* with PM*j* for each resource such as CPU, RAM, disk, network and energy. Besides, additional license fee α for the new VM is applied when horizontal scaling takes place, and is considered as constant (£0.1/hr).

Thus, to get the estimated total cost for VM*x* before and after the action(s) takes place can be given by:

$$\mathrm{VM}x_{Total\_Est\_Cost} = \mathrm{VM}x_{Est\_Cost\_PMi} + \mathrm{VM}x_{Est\_Cost\_PMj} \quad (6.2)$$

6.3 Implementation

The hybrid approach for performance and energy-based cost prediction that aims to integrate the auto-scaling with live migration and estimate the total cost for both migrated and scaled VMs during service operation has been introduced. In order to evaluate this approach, a number of direct experiments have been conducted on the Cloud testbed (see Section 6.3.1) to synthetically generate historical workload data. The process starts by firstly detecting the underloaded and overloaded hosts in order to handle the service performance variation, then predicting the VMs workload using the (***auto.arima***) function in R package [159] to automatically select the best fit model of ARIMA based on AIC or BIC value. Once the VMs workload is predicted, the process goes through the cycle of the approach and considers the correlation between the physical and virtual resources to predict power consumption of the VMs when being run on multiple PMs. Then, the most cost-effective decision(s) is performed, and the total cost is estimated for both migrated and scaled VMs based on their predicted workload and power consumption.

6.3.1 Characterisation of Physical Machines

Four different PMs on the Cloud testbed have been considered. The first three PMs, Host A, C and D, have four core X3430 Intel Xeon CPU, and the last PM, Host B, has an eight-core E3-1230 V2 Intel Xeon CPU. Host A is considered as

the source host and Host B, C and D are considered as the destination's hosts. Host B is the most energy efficient PM, Host C is the similar PM configuration to the source PM (Host A), and Host D is the less energy efficient PM. Also, each PM has a Watt meter [143] attached to directly measure the power consumption. Heterogeneous VMs are created and their monitoring is performed through Zabbix [150], which is also used for resources usage monitoring.

6.4 Experiments and Evaluation

6.4.1 Design of Experiments

A number of direct experiments have been conducted on the Cloud testbed. The overall aim of the experiments is to demonstrate that the hybrid approach for performance and energy-based cost prediction is capable to detect and predict the underloaded/overloaded PMs in order to perform cost-effective decisions. Also, this approach is capable to predict the workload and power consumption as well as estimating the total cost of migrated and scaled VMs when being run on different PMs. Furthermore, the proposed approach focuses on overall cost savings that can be obtained when migrating/scaling the VMs to/on different hosts have different energy characterisation.

Three direct experiments have been conducted for each live migration and auto-scaling operation using three types of VMs with the objective to 1) detect the underloaded and overloaded hosts in order to handle the service performance variation at the PM level; 2) reduce energy-related costs while maintaining performance requirements, 3) identify the most suitable cost-effective decision(s) to handle the service performance variation at the VM level; and 4) estimate the total cost for both migrated and scaled VMs.

To design the experiments, historical data has been generated to represent real workload patterns of Cloud applications (discussed in Section 4.2.1), by using *Stress-ng* tool [73] (see Section 4.4.1) in order to stress all the resources including (CPU, memory, disk and network) on different types of VMs. The generated workload of each VM type has a time interval of four slots (30 minutes each). The first three intervals (slots) are used as the historical data set for prediction, and the last interval (slot) is used as the testing data set to evaluate

the predicted results. A similar approach is used in [160] and followed in this thesis.

6.4.2 Evaluation

6.4.2.1 VMs Workload Prediction

This section presents the quantitative evaluation of the hybrid approach for performance and energy-based cost prediction in terms of VMs live migration and auto-scaling in order to estimate the total cost of VMs during service operation. The figures show the predicted workload results for three types of VMs, small, medium and large, running on multiple PMs based on historical periodic workload pattern.

In Algorithm 6.1, when PM*i* is in the situation of underloaded/overloaded that meets the predefined (lower, upper and max_upper) thresholds, instead of immediately migrating/auto-scaling the VMs, the prediction model is used to minimise the number of VM migration/scaling decisions and avoid unnecessary migration/scaling caused by the small peaks in the workload (false alarm). However, when PM*i* is underloaded which means that the predicted workload is less than or equals to the predefined lower threshold. The proposed Algorithm 6.2 is used to migrate the VMs and re-allocate/allocate them on appropriate PM*j* which has sufficient resources and is potentially more energy efficient, in order to switch PM*i* to power saving mode and hence save energy cost. In the case when PM*i* is overloaded and the predicted workload is in the range of [upper and max_upper threshold]. The proposed Algorithm 6.3 is used to perform the most cost-effective scaling/migration decisions (e.g., resize VMs, migrate existing VMs and resize, or initiate new VMs) and re-allocate/allocate the VMs on the selected PM*j* which has sufficient resources and is potentially more energy efficient.

Figures 6-2, 6-3, and 6-4 depict the results of the migrated and scaled VMs predicted versus the actual workload, including CPU, RAM, disk, and network usage for the VMs. Despite the periodic utilisation peaks, the predicted VMs CPU, RAM and network 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. The predicted

VMs disk workload is also matching the actual workload, but with less accuracy as compared to the CPU, RAM and network prediction results. This can be justified because of the high variations in the generated historical periodic workload pattern of the disk not closely matching in each interval. Besides the predicted VMs' workload mean values, the results also show the high and low 95% and 80% confidence intervals for the predicted workload of each VM based on the ARIMA model.

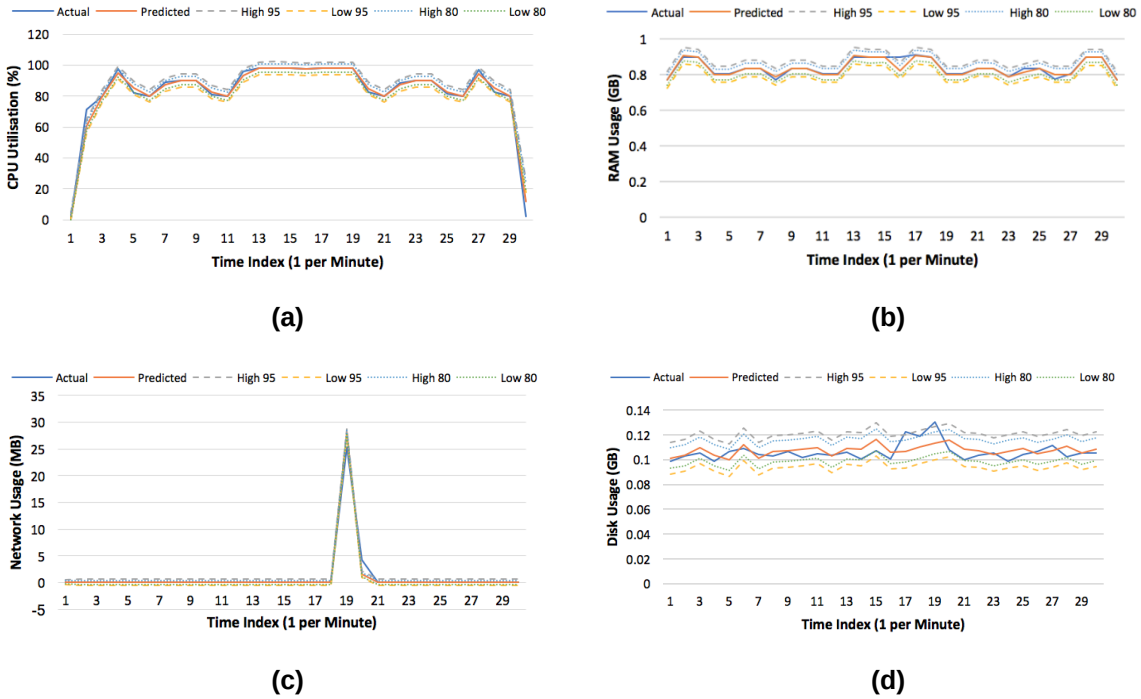


Figure 6-2: The Workload Prediction Results for Small VM.

Table 6-1: Prediction Accuracy for Small VM.

Parameters	ME	RMSE	MAE	MPE	MAPE
CPU Utilisation	0.00486	1.7101	0.5652	-3.4611	4.978
RAM Usage	0.00167	0.0189	0.0055	0.1618	0.6585
Disk Usage	-0.0052	0.1869	0.0461	3.459	6.940
Network Usage	0.00072	0.0051	0.0030	0.64200	2.8612

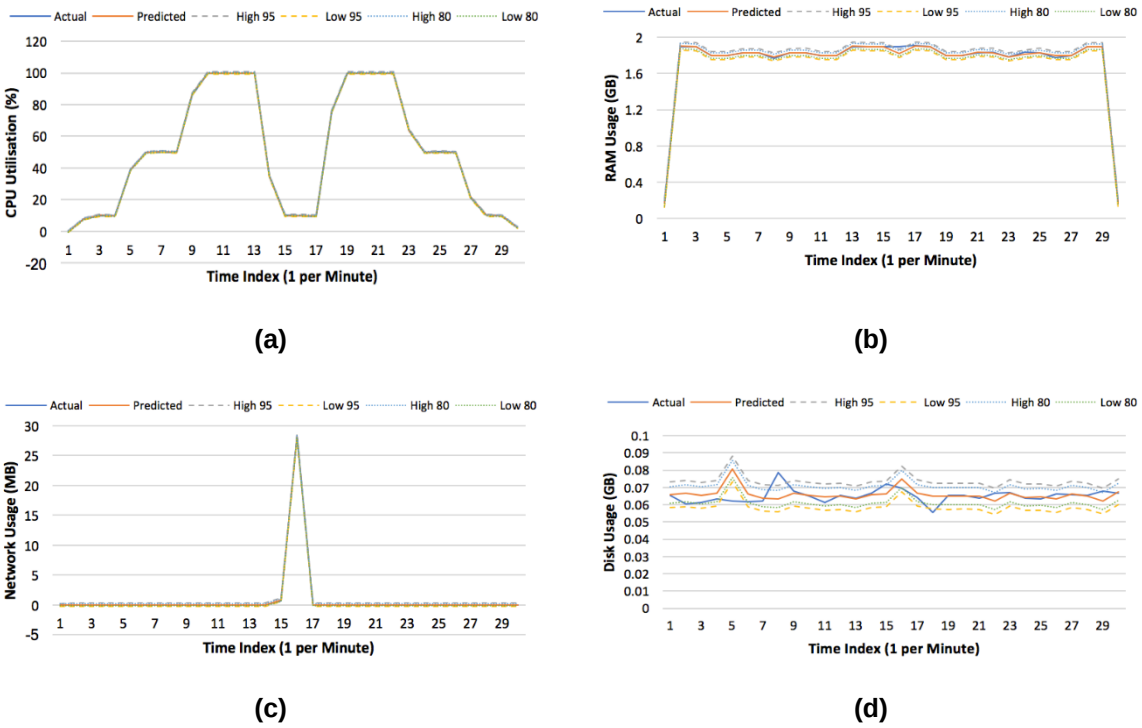


Figure 6-3: The Workload Prediction Results for Medium VM.

Table 6-2: Prediction Accuracy for Medium VM.

Parameters	ME	RMSE	MAE	MPE	MAPE
CPU Utilisation	0.019355	0.2451	0.12275	-3.1443	3.576033
RAM Usage	0.001976	0.0189	0.00588	0.11509	0.333648
Disk Usage	0.000197	0.0940	0.01848	-8.96	9.5482
Network Usage	-0.00005	0.0030	0.00181	-0.2380	2.716369

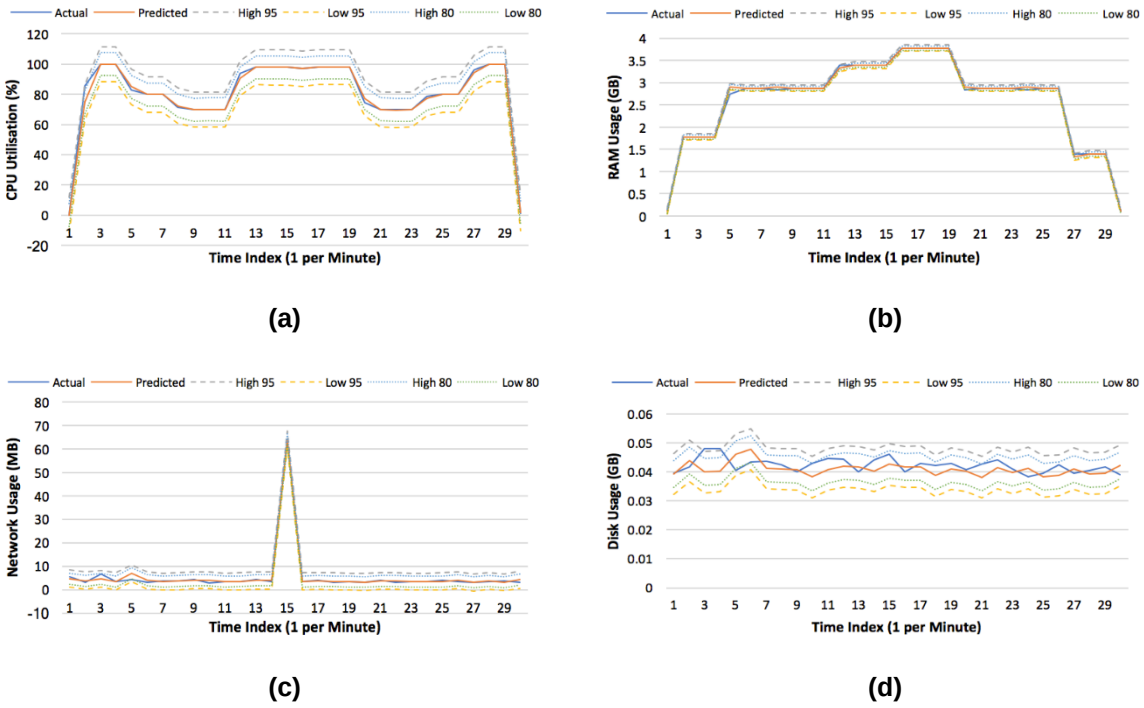


Figure 6-4: The Workload Prediction Results for Large VM.

Table 6-3: Prediction Accuracy for Large VM.

Parameters	ME	RMSE	MAE	MPE	MAPE
CPU Utilisation	0.437240	4.8481	1.39113	0.86261	2.095702
RAM Usage	-0.00097	0.0308	0.00791	-0.0621	0.328699
Disk Usage	-0.08418	1.4943	0.47049	-3.3323	11.57954
Network Usage	-0.00001	0.0028	0.00156	-0.3278	3.637562

In terms of prediction accuracy, a number of metrics have been used to evaluate the results of the predicted workload for three types of VMs, as these metrics have been defined earlier in Section 4.4.2. The accuracy of the predicted VMs workload (CPU, RAM, disk, network) based on a periodic workload is evaluated using these accuracy metrics, as summarised in Tables 6-1, 6-2 and 6-3, respectively.

6.4.2.2 VMs Power Consumption Prediction

Besides the VMs workload prediction, the proposed approach can predict the power consumption for a number of VMs when running on source PM*i* and destination's PM*j* for both live migration and auto-scaling, as described next.

6.4.2.2.1 VMs' Live Migration Power Consumption Prediction

Figures 6-5, 6-6 and 6-7 depict the results of the predicted versus the actual power consumption for a number of VMs running on source PM*i* (Host A) and destination PM*j,* noting that the destination PM*j* can be the most energy efficient PM (Host B), a similar PM configuration to the source PM (Host C) or less energy efficient PM (Host D) comparing to source PM*i*, all of which were based on the migration decision. According to Algorithm 6.2, the migration is performed for the underloaded PM*i* if the selected destination PM*j* has enough resources and does not exceed the upper threshold once the VMs migration takes place.

Also, it is worth mentioning that the predicted power consumption attribution for each VM is affected by the variation in the predicted PM CPU utilisation of all VMs. In terms of prediction accuracy, a number of metrics have been used to evaluate the predicted power consumption for small, medium and large VMs based on a periodic workload pattern as presented in Table 6-4.

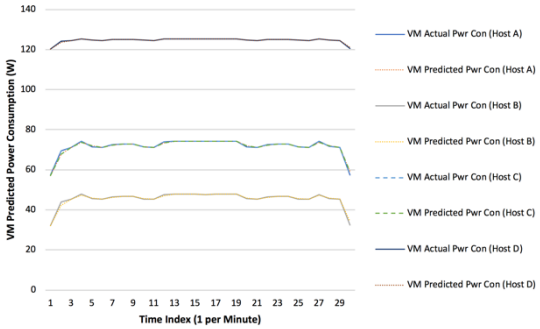


Figure 6-5: Small VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj).

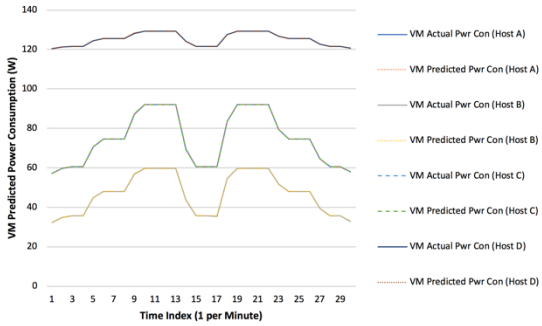


Figure 6-6: Medium VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj).

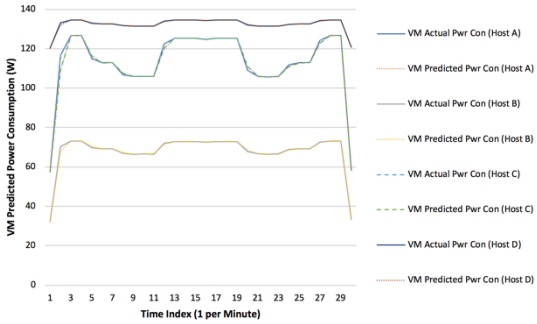


Figure 6-7: Large VM Predicted vs Actual Power Consumption on (Source PMi and Destination PMj).

Table 6-4: Prediction Accuracy for The Predicted Power Consumption for all VMs on Source (Host A) and Destination (Host B, Host C and Host D).

Parameter	VMs	Hosts	ME	RMSE	MAE	MPE	MAPE
VMs Power Consumption	Small VM	Host A	-0.00551665	0.5150904	0.2493285	0.00539324	0.3674461
		Host B	0.005655233	0.4750381	0.2190667	0.04799226	0.5281281
		Host C	-0.00551665	0.5150904	0.2493285	0.00539324	0.3674461
		Host D	0.00246747	0.1537848	0.07028654	0.0023619	0.05689478
	Medium VM	Host A	0.01939327	0.07113483	0.04306951	0.02648363	0.05983904
		Host B	0.01529777	0.05683427	0.03492552	0.03521646	0.08024377
		Host C	0.01939327	0.07113483	0.04306951	0.02648363	0.05983904
		Host D	0.004925887	0.01823638	0.01120869	0.003956332	0.00901164
	Large VM	Host A	-0.2564522	1.533448	0.5685501	-0.2213621	0.5101096
		Host B	-0.07265782	0.5223516	0.193443	-0.0954475	0.2912161
		Host C	-0.2564522	1.533448	0.5685501	-0.2213621	0.5101096
		Host D	0.00000132	0.0000031	0.00000278	0.00000099	0.0000021

6.4.2.2.2 VMs' Auto-scaling Power Consumption Prediction

Figures 6-8 to 6-16 depict the results of the predicted versus the actual power consumption for a number of VMs running on different hosts using different techniques (vertical scaling, *migration and vertically scaling* and horizontal scaling). According to Algorithm 6.3, the vertical scaling is performed for the overloaded VMs on the same host, if the host has enough resources to meet the scaling requirements (vertical scaling is limited to the capacity of PM*i*). Otherwise, the VMs *migration and vertically scaling* or the horizontal scaling are performed for the overloaded VMs on a number of hosts, e.g., (Host B) is the most energy efficient PM, (Host C) has a similar PM configuration as the source PM (Host A), and (Host D) is the less energy efficient PM.

Note that, the vertical scaling was not performed for all types of VMs (e.g., large VM), since the large VM has four CPU cores and the capacity of the hosted PM (e.g., on Host A, Host C or Host D) has the same number of CPU cores as the large VM. Thus, there is no available capacity to perform vertical scaling on the same host. Therefore, only horizontal scaling can be performed with this type of VM. On the other hand, the vertical scaling or *migration and vertically* scaling can be performed for the large VM only on (Host B), since it has eight CPU cores. In terms of prediction accuracy, a number of metrics have been used to evaluate the predicted power consumption for small, medium and large VMs based on a periodic workload pattern as presented in Tables 6-5, 6-6 and 6-7, respectively.

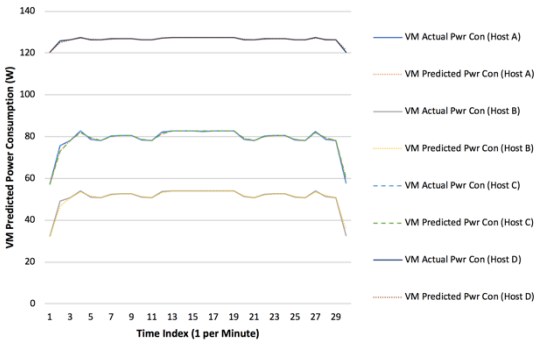


Figure 6-8: Small VM Predicted vs Actual Power Consumption using Vertical Scaling on a Number of PMs.

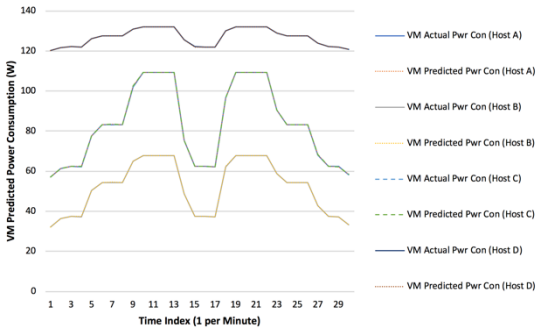


Figure 6-9: Medium VM Predicted vs Actual Power Consumption using Vertical Scaling on a Number of PMs.

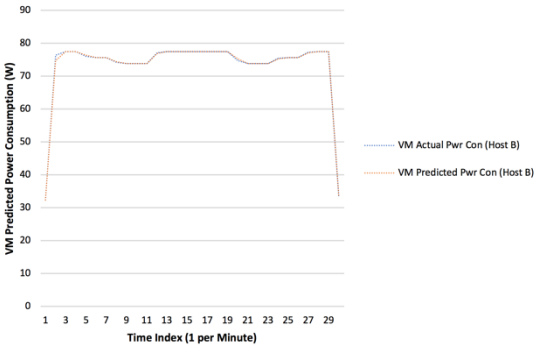


Figure 6-10: Large VM Predicted vs Actual Power Consumption using Vertical Scaling on a Number of PMs.

Table 6-5: Prediction Accuracy for The Predicted Power Consumption for all VMs performs (Vertical Scaling) on Different Hosts.

Parameter	VMs	Hosts	ME	RMSE	MAE	MPE	MAPE
VMs Power Consumption	Small VM	Host A	-0.00827498	0.7726357	0.3739927	0.01611016	0.5129771
		Host B	0.01625062	0.6698005	0.2994985	0.09771351	0.6698014
		Host C	-0.00827498	0.7726357	0.3739927	0.01611016	0.5129771
		Host D	0.00627207	0.2159668	0.09559397	0.00575502	0.07665149
	Medium VM	Host A	0.02908991	0.1067022	0.06460426	0.03629048	0.08216754
		Host B	0.01980407	0.07504576	0.04582536	0.04335952	0.09834909
		Host C	0.02908991	0.1067022	0.06460426	0.03629048	0.08216754
		Host D	0.006596707	0.02445368	0.01499183	0.005249759	0.01192901
	Large VM	Host B	-0.03093269	0.3194241	0.1197874	-0.02547422	0.174785

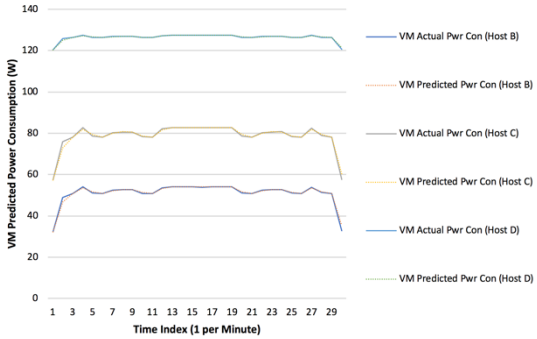


Figure 6-11: Small VM Predicted vs Actual Power Consumption using Migration and Vertically Scaling on a Number of PMs.

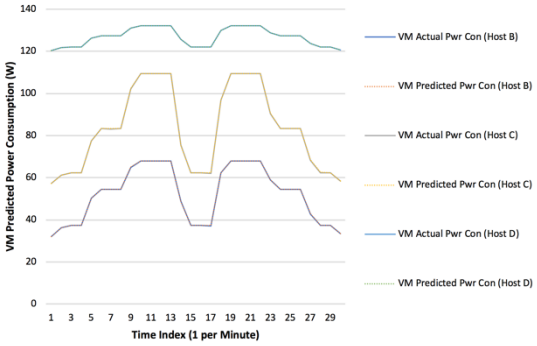


Figure 6-12: Medium VM Predicted vs Actual Power Consumption using Migration and Vertically Scaling on a Number of PMs.

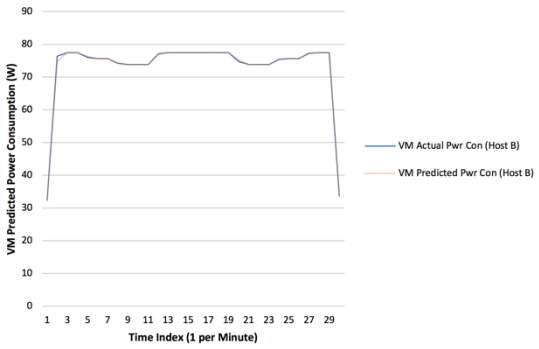


Figure 6-13: Large VM Predicted vs Actual Power Consumption using Migration and Vertically Scaling on a Number of PMs.

Table 6-6: Prediction Accuracy for The Predicted Power Consumption for all VMs performs (Migration and Vertically Scaling) on Different Hosts.

Parameter	VMs	Hosts	ME	RMSE	MAE	MPE	MAPE
VMs Power Consumption	Small VM	Host B	0.01625062	0.6698005	0.2994985	0.09771351	0.6698014
		Host C	-0.00827498	0.7726357	0.3739927	0.01611016	0.5129771
		Host D	0.00627207	0.2159668	0.09559397	0.00575502	0.07665149
	Medium VM	Host B	0.01980407	0.07504576	0.04582536	0.04335952	0.09834909
		Host C	0.02908991	0.1067022	0.06460426	0.03629048	0.08216754
		Host D	0.006596707	0.02445368	0.01499183	0.005249759	0.01192901
	Large VM	Host B	-0.03093269	0.3194241	0.1197874	-0.02547422	0.174785

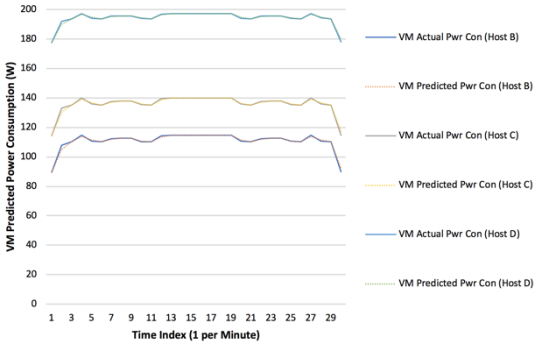


Figure 6-14: Small VM Predicted vs Actual Power Consumption using Horizontal Scaling on a Number of PMs.

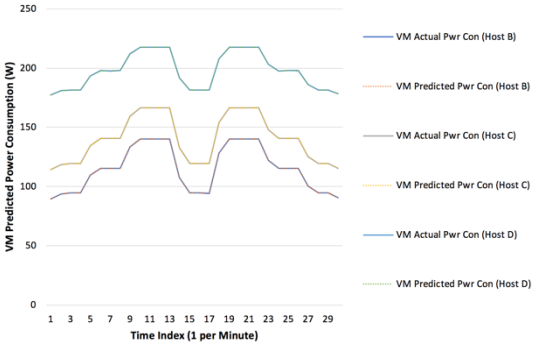


Figure 6-15: Medium VM Predicted vs Actual Power Consumption using Horizontal Scaling on a Number of PMs.

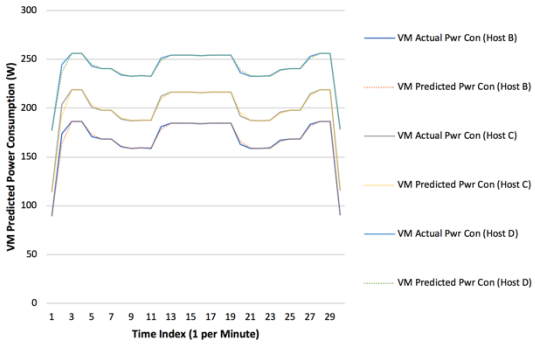


Figure 6-16: Large VM Predicted vs Actual Power Consumption using Horizontal Scaling on a Number of PMs.

Table 6-7: Prediction Accuracy for The Predicted Power Consumption for all VMs performs (Horizontal Scaling) on Different Hosts.

Parameter	VMs	Hosts	ME	RMSE	MAE	MPE	MAPE
VMs Power Consumption	Small VM	Host B	-0.0054593	0.7680777	0.3691814	0.00744378	0.3509133
		Host C	-0.0082749	0.7726357	0.3739927	0.00169475	0.2865351
		Host D	-0.0052990	0.5977032	0.2882975	-0.0005908	0.1515757
	Medium VM	Host B	0.02823278	0.1035619	0.06297163	0.02507514	0.05674791
		Host C	0.02908989	0.1067022	0.06460427	0.02091843	0.04715986
		Host D	0.02224723	0.08158656	0.04950406	0.01128311	0.02533147
	Large VM	Host B	-0.3341869	2.027128	0.7513777	-0.1932349	0.4520876
		Host C	-0.3846782	2.300172	0.8528251	-0.1904874	0.4350779
		Host D	-0.2811938	1.68985	0.6266054	-0.1155186	0.2615967

6.4.2.3 VMs Total Cost Estimation

The proposed approach is also capable of estimating the live migration and auto-scaling total cost for a number of VMs when running on different PMs.

6.4.2.3.1 VMs' Live Migration Cost Estimation

Figure 6-17 shows the results of the estimated total cost for a number of VMs before live migration on (Host A) and after live migration takes place on (Host B, Host C and Host D) along with their migration cost. According to Algorithms 6.1 and 6.2, the migration is performed for the underloaded PM*i* only if the cost of VMs incurred by live migration to the selected destination PM*j* is less than the cost of switching the source PM*i* to power saving mode. In this case, only the small VM can meet these conditions to be migrated to the selected destination PM*j*, if it is the only VM running on the source PM*i*.

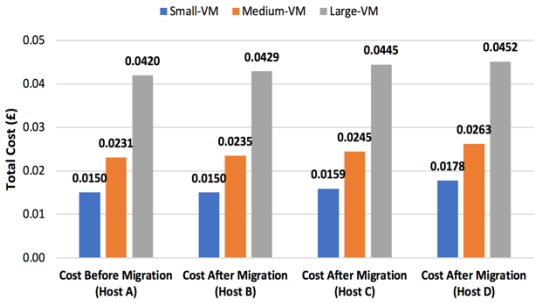


Figure 6-17: Estimated Total Cost Before vs After Migration on Different PMs.

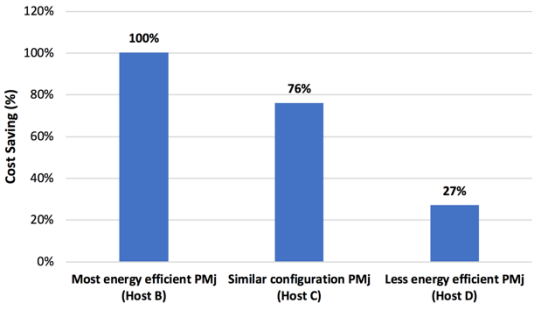


Figure 6-18: Estimated Cost Saving for Migrating Small VM to Different PMs.

In addition, Figure 6-18 shows the results of the estimated cost saving that can be achieved for the small VM when being migrated to different hosts. This comes at the cost of the power savings that are gained by switching the source (Host A) to power saving mode. For example, when the VM is migrated to the most energy efficient PM (Host B), it can achieve approximately 100% cost saving which means the cost that can be saved by switching PM*i* to power saving mode (idle state) minus the cost incurred by the migration decision. With a similar PM configuration to the source (Host C), it can achieve around 76% cost saving and with the less energy efficient PM (Host D), it can achieve about 27%.

Further, the energy efficiency of the hosts plays an important role to reduce the overall energy consumption. Thus, selecting the appropriate hosts to

migrate the VMs have a significant impact on the overall cost saving (e.g., migrating the VMs to most energy efficient PM (Host B) can be more cost-effective than migrating the VMs to less energy efficient PM (Host D)).

6.4.2.3.2 VMs' Auto-Scaling/Migration Cost Estimation

Figures 6-19, 6-20 and 6-21 show the results of the estimated total cost for three types of VMs running on a number of PMs using different scaling/migration strategies. According to Algorithm 6.3, the vertical scaling is performed on the same host, if the host has enough resources to meet the scaling requirements. Otherwise, the VMs *migration and vertically scaling* or the horizontal scaling are performed on a number of hosts, e.g., (Host B) is the most energy efficient PM, (Host C) is a similar PM configuration to the source PM (Host A), and (Host D) is the less energy efficient PM.

As mentioned earlier, the vertical scaling was not performed with the large VM, since it has the same number of CPU cores as the hosted PM (e.g., on Host A, Host C or Host D), which means that the host is fully utilised via the VM. However, the vertical scaling can be performed for the large VM only on (Host B) that has eight cores, as shown in Figures 6-19 and 6-20, respectively.

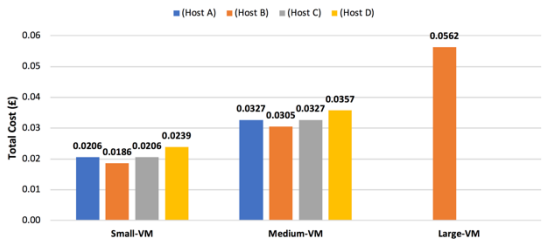


Figure 6-19: Estimated Vertical Scaling VMs Total Cost.

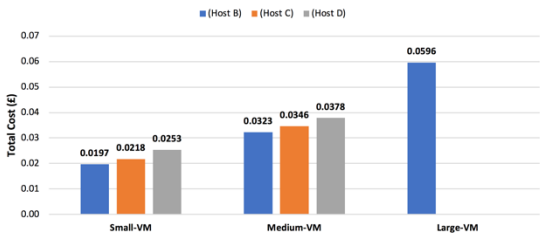


Figure 6-20: Estimated Migration and Vertically Scaling VMs Total Cost.

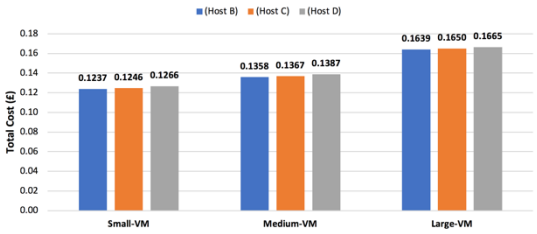


Figure 6-21: Estimated Horizontal Scaling VMs Total Cost.

Choosing between different scaling strategies can have a significant impact on the cost of the scaled VMs (e.g., vertical scaling can be more cost-effective than the proposed *migration and vertically scaling* or the horizontal scaling when the VMs are scaled on a similar host configuration), as shown in Figures 6-19, 6-20 and 6-21, respectively. This can be justified because vertical scaling has no additional costs in terms of migration cost (e.g., in the case of *migration and vertically scaling*) or software license for new VMs [18] (e.g., in the case of horizontal scaling). However, the vertical scaling technique is limited to the capacity of the host [18], [102]. Therefore, the proposed *migration and vertically scaling* mechanism can help to select the most suitable cost-effective scaling strategy, rather than just only choosing between scaling up/out.

As shown in Figures 6-20 and 6-21, the proposed *migration and vertically scaling* mechanism outperforms the horizontal scaling one. This can be justified because of the additional cost in terms of software license for the new VMs when horizontal scaling is performed is higher than the cost of live migration for the VMs when *migration and vertically scaling* is performed. Furthermore, selecting the appropriate hosts in terms of their energy efficiency to scale the VMs have a significant impact on the total cost of the scaled VMs (e.g., horizontal scaling using most energy efficient PM can be more cost-effective than horizontal scaling when using less energy efficient PM).

6.5 Summary

This chapter has presented and evaluated a new hybrid approach for performance and energy-based cost prediction. This approach dynamically supports decision-making regarding auto-scaling and live migration costs while at the same time being aware of the impact on the energy consumption and performance of the application during service operation. This hybrid approach integrates auto-scaling with live migration in order to estimate the total cost of heterogeneous VMs by considering their resource usage and power consumption, while at the same time maintaining the expected level of application performance. The results show that the proposed hybrid approach can detect the underloaded and overloaded hosts in order to perform the most cost-effective decision(s) to handle the service performance variation. It can also predict the workload, power consumption and estimate the total cost for both

migrated and scaled VMs when being run on different PMs, with a high prediction accuracy based on historical workload patterns.

Chapter 7. Conclusion

This chapter concludes this thesis and provides a summary of the conducted research, as presented in Section 7.1. This is followed by an overall results discussion of the conducted experiments in Chapters 3, 4, 5 and 6, along with a comparison of the related work with the work introduced in this thesis, as presented in Section 7.2. The key contributions of the research are provided and discussed in Section 7.3. This is followed by a discussion on the limitations of the research based on the results obtained from the conducted experiments and the comparison with the related work, as presented in Section 7.4. Finally, future work directions that can be explored based on the work presented in this research are suggested and discussed in Section 7.5.

7.1 Research Summary

With the wide adoption of Cloud Computing, Cloud providers consider energy consumption as one of the biggest cost factors to be maintained within their infrastructures [1]–[3]. Consequently, modelling a new cost mechanism for Cloud services that can be adjusted to the energy costs has increasingly become an important research topic for both academia and industry, as presented in Chapter 3 and Chapter 4. Further, Cloud Computing can be used to obtain cost benefits through *proactive* efficient resource management techniques such as VMs consolidation and resource provisioning. These techniques can help Cloud providers to make enhanced cost decisions in terms of reducing energy-related costs while maintaining performance requirements. Consequently, estimating the future cost of Cloud services can help the service providers offer suitable services that meet their customers' requirements, as presented in Chapter 5 and Chapter 6.

Therefore, the work presented in this thesis aims at enabling the awareness of energy consumption, performance variation and cost in a Cloud environment. A cloud system architecture is introduced along with the main component *Cost Modeller* to fulfil this aim. Firstly, an energy-based cost model is developed to attribute the PM's energy consumption to VMs and measures the actual resource usage, power consumption and the total cost for each VM. Then,

the energy-based cost prediction framework is introduced to predict workload, power consumption and estimate the total cost of the VMs. Finally, a performance and energy-based cost prediction framework is introduced to combine VMs consolidation and resource provisioning in order to design cost-effective strategies, while taking into consideration the trade-off between cost, energy efficiency and performance variation of Cloud services.

- **Chapter 2:** presents the essential background and reviews the literature on the subject of the energy-related cost issues, prediction models and resource management in Cloud Computing. It starts by introducing the fundamental concepts of Cloud Computing with a detailed description of its definition, system architecture, services types, deployment types and virtualisation technologies. The aspects of Cloud applications and their workload patterns, as well as related benchmarks, are discussed. A description of Cloud Computing pricing models is presented. This is followed by positioning the work in the relevant literature, focusing on the energy-related cost issues, prediction models and resource management in Cloud Computing. The energy-related cost issues are highlighted, along with a detailed discussion of the closely related work. It then discusses the prediction models related to the workload, energy consumption and cost of Cloud services. A discussion of the closely related work is also presented. Finally, it reviews the existing work on dynamic resource management, including VMs consolidation and resource provisioning, along with a discussion of the closely related work and the thesis scope.
- **Chapter 3:** introduces the system architecture that supports energy, performance and cost awareness of Cloud infrastructure services. Detailed descriptions of the proposed system architecture main components along with their roles and how they interact with the proposed component *Cost Modeller* to achieve their objectives are discussed. This is followed by presenting an energy-based cost model that considers energy consumption as a key parameter. This model focuses on fairly attributing the PM's energy consumption to heterogeneous VMs based on their vCPU utilisation and size. Then, measures the actual resource usage, power consumption and the total cost for each VM. A thorough

discussion of the development of this model is provided. Early experiments along with their results are performed to evaluate the ability of the proposed system architecture along with the proposed model in terms of supporting cost and energy awareness at the VM level in a Cloud environment.

- **Chapter 4:** introduces an energy-based cost prediction framework used to enable cost and energy awareness at the VM level in a Cloud environment. This framework focuses on predicting the VMs' workload based on historical workload patterns and correlating the predicted VMs workload with physical resources to predict the power consumption of the VMs. Then, estimate the total cost of the VMs' during service operation. A thorough discussion of the development of this framework is provided. A number of direct experiments on the Cloud testbed are demonstrated along with their results to evaluate the proposed framework in terms of its capability to estimate the total cost of VMs by considering their resource usage and power consumption during service operation.
- **Chapter 5:** introduces the performance and energy-based cost prediction framework used the prediction framework presented in Chapter 4 for enabling energy, performance and cost awareness of Cloud infrastructure services. This framework focuses on enhancing VMs consolidation and resource provisioning techniques in order to design cost-effective strategies and prevent performance loss at different levels. A thorough discussion of the development of this framework is provided. A number of direct experiments on the Cloud testbed are demonstrated along with their results to evaluate the capability of the proposed framework to estimate the live migration and auto-scaling total cost for heterogeneous VMs at service operation.
- **Chapter 6:** presents a hybrid approach for the presented framework in Chapter 5 by integrating auto-scaling with live migration in order to estimate the total cost of VMs by considering their resource usage and power consumption. This framework focuses on detecting the underloaded and overloaded hosts in order to perform the most cost-effective decision(s) to handle the service performance variation. A thorough discussion of the development of this approach is provided. A

number of direct experiments on the Cloud testbed are demonstrated along with their results to evaluate the capability of the hybrid approach to estimate the total cost for heterogeneous VMs incurred by different decisions at service operation.

7.2 Research Outcomes

A cloud system architecture is introduced in this thesis to enable the awareness of energy consumption, performance variation and total cost of Cloud infrastructure services. The *Cost Modeller* is the main architectural component including the other contributions of this thesis are highlighted below.

7.2.1 Energy-based Cost Model

Chapter 3 has investigated how the cost models of Cloud services can be identified in a Cloud environment. In this regard, energy consumption considered as one of the important parameters that influence the cost of Cloud services. The power consumption at the PM level can be easily identified but is not directly measured at the VM level. Thus, identifying how the physical resources are correlated with the virtual resource's usage and their impact on energy consumption is important for Cloud service providers.

The conducted experiments on a Cloud testbed have shown an early evaluation of the ability of the proposed system architecture in terms of supporting cost and energy awareness at the VM level, which addresses the first research question (**Q.1** – see Section 1.2).

The overall results show that the proposed energy-based cost model can fairly attribute the PM's energy consumption to the VMs and measure the actual resource usage, power consumption and the total cost for a number of VMs, as presented in Section 3.6.2. Unlike other existing works, this approach considers the heterogeneity of the VMs, with respect to the actual resource usage, power consumption and the total cost. These VMs also runs on two PMs having different characteristics with different energy consumption.

Furthermore, the experiments have shown that the measured total cost for the same type of VMs when being run on Host B is less than the total cost when being run on Host A, since Host B has less power characteristics in terms of the idle and active as compared to Host A, as presented in Section 3.6.2. Hence, enabling cost and energy awareness at the VM level can help Cloud service providers to make enhanced cost decisions and efficiently manage their resources.

7.2.2 Energy-based Cost Prediction Framework

Chapter 4 has presented prediction methods along with developed mathematical modelling. The aim of the proposed energy-based cost prediction framework is to address the second research question as stated in (**Q.2 –** see Section 1.2) by predicting the workload, power consumption and estimating the total cost of heterogeneous VMs during service operation based on historical workload pattern. A number of direct experiments were conducted on a Cloud testbed to evaluate the capability of the prediction models.

The overall results show that the proposed framework can attribute the PM's energy consumption to the VMs and predict the resource usage, power consumption and estimate the total cost for the VMs with a high prediction accuracy based on Cloud workload patterns, as presented in Section 4.4.2. Unlike other existing works, this framework considers the heterogeneity of the VMs, with respect to predict resource usage, power consumption and estimate the total cost. Besides the prediction, these VMs also runs on two PMs having different characteristics with different workloads.

Furthermore, the experiments have shown that the estimated cost for the same type of VMs when being run on Host B is less than the estimated cost when being run on Host A, since Host B has less power characteristics as compared to Host A. Hence, enabling cost and energy awareness at the VM level can help Cloud service providers to make enhanced cost decisions and efficiently manage their resources, leading towards a reduction of energy consumption, and therefore lowering the operational costs for Cloud providers. Furthermore, estimating the future cost of Cloud services can help service providers offer suitable services that meet their customers' requirements.

Despite the high variation of the workload utilisation, the accuracy metrics indicate that the predicted VMs workload and power consumption achieve high prediction accuracy along with the estimated total cost.

7.2.3 Performance and Energy-based Cost Prediction Framework

Chapter 5 has investigated the issues related to VMs consolidation and resource provisioning in a Cloud environment in terms of performance variation and energy consumption. Thus, understanding the impact of VMs consolidation and resource provisioning is essential to design cost-effective strategies for Cloud services. A set of algorithms that deal with VMs consolidation and resource provisioning are proposed with the aim to minimise the overall costs incurred by the performed decisions.

The aim of the proposed performance and energy-based cost prediction framework is to address the third and fourth research questions as stated in (**Q.3** and **Q.4** – see Section 1.2) by estimating the total cost of the migrated and scaled VMs, considering their resource usage and power consumption, while maintaining the expected level of service performance. A number of direct experiments were conducted on a Cloud testbed to evaluate the capability of the prediction models.

The overall results show that the proposed framework can predict the workload, power consumption and estimate the total cost for a number of VMs when being run on multiple PMs using different migration and scaling strategies, as presented in Section 5.4.2. The experiments have shown that the estimated migration cost for the same type of VMs when being migrated to Host B is less than the estimated migration cost when being migrated to Host D, since Host B is more energy efficient as compared to Host D. Hence, the estimated migration cost recovery can be achieved when the migration is performed to the most energy efficient Host B. However, when the migration is performed to the less energy efficient Host D, only a large VM can recover the migration cost. Moreover, choosing between different scaling strategies (vertical and horizontal scaling) can have a significant impact on the cost of the scaled VMs. For example, when performed vertical scaling on the same host, Host A, it can be more cost-effective than performed horizontal scaling on a similar host

configuration, Host C. Also, horizontal scaling on the most energy efficient host, Host B, can be more cost-effective than horizontal scaling when using less energy efficient host, Host D. In addition, the results have shown that the proposed self-configuration auto-scaling mechanism outperforms the predefined one, since the predicted power consumption and cost are lower. Thus, these mechanisms can help Cloud providers to make enhanced cost decisions in terms of selecting the most cost-effective decision for both live migration and auto-scaling techniques.

7.2.4 A Hybrid Approach for Performance and Energy-based Cost Prediction

Chapter 6 has integrated the VMs auto-scaling with dynamic VMs allocation into a hybrid approach in this research context. A set of algorithms that detect the underloaded and overloaded hosts in order to perform the most cost-effective decision(s) are proposed with the aim to minimise the overall costs incurred by the performed decisions.

The aim of the proposed hybrid approach for the performance and energy-based cost prediction is to address the fifth research question as stated in (**Q.5** – see Section 1.2) by integrating auto-scaling with live migration to handle the service performance variation. Then, predict the workload, power consumption and estimate the total cost for both migrated and scaled VMs during service operation based on historical workload data. A number of direct experiments were conducted on a Cloud testbed to evaluate the capability of the hybrid approach to estimate the total cost for heterogeneous VMs incurred by different decisions at service operation.

The overall results show that the proposed approach can detect the underloaded and overloaded hosts in order to perform the most cost-effective decision(s) and predict the workload, power consumption as well as estimate the total cost for a number of VMs when being run on multiple PMs using different migration and scaling strategies, as presented in Section 6.4.2.

The experiments have shown that in the case of the underloaded host the estimated cost saving of the VMs can be achieved only if the cost of VMs incurred by live migration to the selected destination PM*j* is less than the cost of switching

the source PM*i* to the power saving mode. In this regard, the energy efficiency of the selected hosts plays an important role to reduce the overall energy consumption. Thus, selecting the appropriate hosts to migrate the VMs has a significant impact on the overall cost saving that can be achieved. In the case of the overloaded host, choosing between different scaling/migration strategies have a significant impact on the total cost of the VMs. Furthermore, selecting the appropriate hosts in terms of their energy efficiency to scale the VMs have a significant impact on the total cost of the VMs. Thus, the proposed approach can help Cloud providers to make enhanced cost decisions in terms of selecting the most suitable cost-effective migration and scaling strategies.

7.2.5 Comparison of Research Approaches with Related Work

Enabling the awareness of energy consumption, performance variation and cost at the virtual level in Cloud environments has become significant and attracted the attention of many researchers. As discussed in Section 2.5, different approaches and models have been introduced to identify cost and energy consumption for VMs in a Cloud environment. Table 7-1 presents a comparison of these related cost and energy models along with the models introduced in this thesis for modelling a new cost mechanism for Cloud services that can be adjusted to the actual energy costs.

Table 7-1: Comparison of Cost and Energy Models.

Criteria / by	Cost Model based on VMs Resource Utilisation Consideration	Actual Power Consumption Consideration	
		PMs level	VMs level
Belli et al. [87]	Homogeneous VMs only.	Not considered.	Not considered.
Jin et al. [81]	Homogeneous VMs only.	Not considered.	Not considered.
Berndt and Maier [23]	Homogeneous VMs only.	Not considered.	Not considered.
Mao and Humphrey [17]	Homogeneous and heterogeneous VMs.	Not considered.	Not considered.
Chard et al. [89]	Homogeneous and heterogeneous VMs.	Not considered.	Not considered.
Yousefipour et al. [93]	Homogeneous and heterogeneous VMs.	Homogeneous PMs only.	Not considered.
Jung et al. [12]	Homogeneous VMs only.	Homogeneous PMs only.	Not considered.
Hinz et al. [10]	Homogeneous and heterogeneous VMs.	Homogeneous PMs only.	Homogeneous and heterogeneous VMs, but only based on the number of allocated virtual CPUs to each VM.
This Research	Homogeneous and heterogeneous VMs.	Homogeneous and heterogeneous PMs.	Homogeneous and heterogeneous VMs, based on the number and the actual utilisation of the virtual CPUs assigned to each VM.

As shown in Table 7-1, most of the related work [87], [81], [23], [17], [89] aimed to improve the cost efficiency in Cloud environments in order to meet the performance requirements, customers' demands and efficient resource utilisation, but do not consider the energy consumption of the resources. The other models, presented in [93], [12] considered the energy consumption, but their focus is only at the physical level in order to consolidate the VMs and minimise the number of active hosts. The only exception is the model presented in [10] which considers the energy consumption at both physical and virtual levels, though this is still limited as their model only considers the number of allocated virtual CPUs to each VM without consideration of the actual utilisation.

The energy-based cost model presented in this thesis is different when compared to existing models found in the literature. It considers attributing the PM's idle and active power consumption to heterogeneous VMs based on their vCPU utilisation and size, as discussed in Chapter 3. Thus, the model introduced in this research is unique as it considers the heterogeneity of the VMs along with their actual resource usage, power consumption, and the total cost.

In terms of estimating the future cost of VMs during the service operation, it would first require predicting their workload, which can be then translated into energy based on their physical resource usage. Table 7-2 presents a comparison of the related work along with the work presented in this thesis for predicting.

Table 7-2: Comparison of Prediction Approaches.

Criteria / by	Workload Prediction Consideration		Energy Prediction Consideration		Cost Estimation Consideration
	PMs level	VMs level	PMs level	VMs level	
Gong et al. [97], Huang et al. [99]	Homogeneous PMs only.	Homogeneous VMs only.	Not considered.	Not considered.	Not considered.
Farahnakian et al. [100]	Homogeneous and heterogeneous PMs.	Homogeneous and heterogeneous VMs.	Not considered.	Not considered.	Not considered.
Zhang et al. [20]	Not considered.	Heterogeneous VMs.	Not considered.	Not considered.	Not considered.
Fang et al. [101]	Homogeneous PMs only.	Not considered.	Not considered.	Not considered.	Not considered.
Yang et al. [102], [18]	Not considered.	Heterogeneous VMs.	Not considered.	Not considered.	Not considered.
Smith et al. [105]	Not considered.	Not considered.	Homogeneous PMs only.	Not considered.	Not considered.
Kistowski et al. [106]	Not considered.	Not considered.	Heterogeneous PMs.	Not considered.	Not considered.

Li et al. [48]	Not considered.	Not considered.	Homogeneous PMs only.	Homogeneous VMs only.	Not considered.
Farahnakian et al. [108]	Heterogeneous PMs.	Not considered.	Heterogeneous PMs.	Not considered.	Not considered.
Subirats and Guitart [109]	Heterogeneous PMs.	Homogeneous VMs only.	Heterogeneous PMs.	Homogeneous VMs only.	Not considered.
Jiang et al. [19]	Heterogeneous PMs.	Heterogeneous VMs.	Not considered.	Not considered.	Based on the resource usage.
Roy et al. [110]	Not considered.	Homogeneous VMs only.	Not considered.	Not considered.	Based on the resource usage.
Sharma et al. [111]	Heterogeneous PMs.	Homogeneous VMs only.	Not considered.	Not considered.	Based on the resource usage.
Liu et al. [112]	Homogeneous PMs only.	Homogeneous VMs only.	Homogeneous PMs only.	Homogeneous VMs only.	Based on the resource usage and power consumption cost for homogeneous PMs and VMs.
This Research	Homogeneous and heterogeneous PMs.	Homogeneous and heterogeneous VMs.	Homogeneous and heterogeneous PMs.	Homogeneous and heterogeneous VMs.	Based on the resource usage and power consumption cost for homogeneous/heterogeneous PMs and VMs.

As discussed in Section 2.6, most of the related approaches presented in [18], [20], [97], [99]–[102] aimed at predicting the workload in order to improve resource utilisation in Cloud environments, yet not considering the energy consumption of the predicted workloads. The other approaches, presented in [105], [106], [108] considered the prediction of energy consumption, but these approaches only take into account the prediction of the power consumption at PMs level and do not consider the prediction at VMs level. However, the work presented in [48], [109] considered predicting energy consumption at both physical and virtual levels. This work is still limited as the model in [48] assumed that all the PMs and VMs are homogeneous, whereas, the model in [109] only considers a linear relationship between the CPU utilisation and the energy consumption in order to predict the power at the VMs level. Further, the work presented in [19], [110], [111] considered the prediction of workload and the estimation of cost for the VMs, but not the energy consumption which would influence the overall cost estimation of Cloud services.

As shown in Table 7-2, the work presented in [112] is the only work that has a similar approach to the one introduced in this thesis in terms of the prediction of the workload and energy consumption as well as the estimation of VMs cost. Nonetheless, their approach does not consider the heterogeneity of the PMs or the VMs, whereas the prediction approach introduced in this thesis takes into account the heterogeneity of the PMs and the VMs.

The energy-based cost prediction framework presented in this thesis first predicts 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. After that, the total cost of VMs is estimated based on their predicted workload and power consumption, as discussed in Chapter 4.

Additionally, Cloud service providers implement dynamic resource management through VMs' consolidation and resource provisioning techniques in order to meet the performance requirements of applications, while minimising the operation costs and energy consumptions in Cloud data centres. Section 2.7 has reviewed the related work on VMs' consolidation and resource provisioning mechanisms in Cloud environments. The following Table 7-3 presents a comparison of the closely related work on the prediction models for VMs' consolidation and resource provisioning that considers the workload, energy consumption and cost in Cloud environments, along with the work presented in this thesis.

Table 7-3: Comparison of Prediction Models for VMs' Consolidation and Resource Provisioning.

Criteria / by	Workload Prediction Consideration		Energy Prediction Consideration		Cost Estimation Consideration	
	PMs level	VMs level	PMs level	VMs level	Cost of Migration	Cost of Scaling
Farahnakian et al. [118]	Heterogeneous PMs.	Heterogeneous VMs.	Not considered.	Not considered.	Not considered.	—
Beloglazov and Buyya [123]	Homogeneous PMs only.	Not considered.	Not considered.	Not considered.	Not considered.	—
Zhou et al. [16]	Heterogeneous PMs.	Not considered.	Heterogeneous PMs.	Not considered.	Considered the cost of migration.	—
Dawoud et al. [137]	Homogeneous PMs only.	Homogeneous VMs only.	Not considered.	Not considered.	—	Not Considered.
Meng et al. [138]	Homogeneous PMs only.	Homogeneous VMs only.	Not considered.	Not considered.	Not considered.	—
Dutta et al. [14]	Homogeneous PMs only.	Homogeneous and heterogeneous VMs.	Not considered.	Not considered.	—	Considered the cost of the horizontal and vertical scaling.
This Research	Homogeneous and heterogeneous PMs.	Homogeneous and heterogeneous VMs.	Homogeneous and heterogeneous PMs.	Homogeneous and heterogeneous VMs.	Considered the cost of migration and their recover cost.	Considered the cost of the horizontal and vertical scaling.

In terms of VMs consolidation, the work presented in [118], [123], [16] employed workload prediction models based on historical data to avoid unnecessary VM migrations and minimise energy consumption and SLA violations. These models focused on improving the performance of Cloud applications by reducing the number of overloaded hosts, but without explicitly considering energy and cost of VMs migrations, as a part of VMs consolidation decision criterion.

In terms of VMs resource provisioning, the work presented in [137], [138], [14] considered the prediction of resources provisioning to handle future workload demand while maintaining the SLOs, but these approaches do not consider the power consumption of required resources incurred due to scaling decisions.

The performance and energy-based cost prediction framework along with the hybrid approach presented in this thesis dynamically supports VMs live migration and auto-scaling decisions, considering the trade-off between cost, power consumption, and performance during service operation. This work detects the underloaded and overloaded hosts in order to perform the most cost-effective decision(s) to handle service performance variations. Also, it predicts the workload, power consumption as well as estimates the total cost for a number of VMs when being run on multiple PMs using different migration and scaling strategies, as discussed in Chapters 5 and 6, respectively.

7.3 Research Contributions

In order to address the research questions of this thesis (see Section 1.2), a number of contributions have been presented in this thesis and they are mainly summarised as follows:

- *A Cloud system architecture*. This architecture has been proposed along with the main component in this research *Cost Modeller* in order to enable the awareness of energy consumption, performance variation and cost in a Cloud environment. *An energy-based cost model* is introduced to address the first research question (**Q.1**) by enabling cost and energy-awareness at the VM level. The results presented in Chapter 3 show that the proposed energy-based cost model can fairly attribute the PM's

energy consumption to heterogeneous VMs and measures the actual resource usage, power consumption and the total cost for each VM.

- *An energy-based cost prediction framework*. A number of models have been introduced within this framework with the overall objective to address the second research question (**Q.2**) by predicting the workload, power consumption and estimating the total cost of the VMs during service operation. Firstly, the VMs' workload is predicted using the ARIMA model based on historical periodic workload patterns. Then, the predicted VM workload is correlated with the physical resources using regression models introduced within this framework in order to predict the PM power consumption, from which the predicted VMs power consumption is identified. After that, the total cost is estimated based on the predicted workload and power consumption for each VMs'. The results presented in Chapter 4 show that a high prediction accuracy of the VMs' workload and power consumption along with their estimated total cost has been achieved by the introduced framework.
- *A performance and energy-based cost prediction framework*. A number of models and algorithms have been introduced within this framework with the overall objective to address the third and fourth research questions (**Q.3** and **Q.4**) by estimating the total cost of heterogeneous VMs, considering their resource usage and power consumption, while maintaining the expected level of application performance. This framework introduced two approaches that can be used for VMs consolidation and resource provisioning in order to design cost-effective strategies and prevent performance loss at different levels. This framework works by predicting the workload, power consumption and estimating the total cost of the migrated and scaled VMs during service operation based on historical workload data. The results presented in Chapter 5 show the capability of the proposed framework to estimate the live migration and auto-scaling total cost for heterogeneous VMs at service operation.
- *A hybrid approach for performance and energy-based cost prediction*. A number of models and algorithms have been introduced within this approach with the overall objective to address the fifth research question

(**Q.5**) by integrating auto-scaling with live migration in order to estimate the total cost of heterogeneous VMs, considering their resource usage and power consumption, while maintaining the expected level of application performance. This approach works by detecting the underloaded and overloaded hosts in order to perform the most cost-effective decision(s) to handle the service performance variation. The results presented in Chapter 6 show the capability of this hybrid approach to predict the workload, power consumption and estimate the total cost of the performed decisions.

7.4 Limitations

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

- The proposed energy-based cost prediction framework only considers the CPU utilisation, (PM CPU utilisation and number of vCPUs assigned for each VM), when modelling and predicting the energy to the VMs. Other resources such as memory, disk and network are not taken into consideration. However, many of the related work concluded that the CPU utilisation is highly correlated with the power consumption. Thus, the CPU is the only resource that affects the power consumption and any other resources do not have any impact on the power, or indirectly impact on the power, driven only through the CPU utilisation.
- The VM workload prediction in the proposed energy-based cost prediction framework is based only on historical periodic workload pattern. Additional Cloud applications workload patterns (e.g., unpredictable, once-in-a-lifetime, and continuously changing), can be further considered to broaden the scope of using the proposed framework to predict the workload, power consumption and estimate the total cost of the VMs based on different types of workload patterns.

- The proposed performance and energy-based cost prediction framework only considers the CPU utilisation and memory usage as thresholds to predict the service performance variation and trigger the appropriate action(s) accordingly. Since the direct experiments are conducted on a local Cloud testbed, thus there is no impact on disk or network in terms of resources competition. However, the performance prediction algorithms of the proposed framework could be extended by considering such resources (disk and network) when identifying the thresholds.

7.5 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-based cost prediction framework presented in this thesis predicts the PMs power consumption based on the correlation of the predicted PM CPU utilisation with PM power consumption using regression models. Then, fairly attribute the predicted PM power consumption to heterogeneous VMs based on the allocated vCPUs and their utilisation by each VM. An extension to this is to also consider the performance counters to predict the subsystems power consumption including memory, disk, and network. This would be a beneficial enhancement which may increase the accuracy of the predicted power consumption at the VM level.
- The workload prediction within the proposed framework is using a time series prediction model (an ARIMA model) based on historical periodic workload patterns. Another suggested extension in terms of prediction is to consider more powerful data-driven methods, e.g., Machine Learning (ML) techniques including an Artificial Neural Network (ANN) or a Deep Neural Network (DNN) with 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, power consumption and estimate the total cost of the VMs based on different prediction techniques along with different types of workload patterns.

- The work presented in this thesis considered the heterogeneity as a different number of resources with different CPUs architectures. Nowadays, heterogeneity refers to different hardware with different architectures that may contain accelerators (e.g., Graphic Processing Units (GPUs) and Field Programmable Gate Arrays (FPGAs)). Since the majority of the proposed energy models in the literature are based on the CPU utilisation, a promising extension of the proposed work is to consider the impact of hardware accelerators, such as GPUs and FPGAs on the energy consumption and service performance. This extension would be useful when modelling and identifying the energy consumption and total cost of Cloud services.

References

[1] T. Mukherjee, K. Dasgupta, G. Jung, and H. Lee, “An Economic Model for Green Cloud,” in *MGC ’12 Proceedings of the 10th International Workshop on Middleware for Grids, Clouds and e-Science*, 2012.

[2] X. Zhang, J. Lu, and X. Qin, “BFEPM: Best fit energy prediction modeling based on CPU utilization,” in *Proceedings - 2013 IEEE 8th International Conference on Networking, Architecture and Storage, NAS 2013*, 2013, pp. 41–49.

[3] 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.

[4] M. Bagein *et al.*, “Energy Efficiency for Ultrascale Systems : Challenges and Trends from Nesus Project 1 . Heterogeneous infrastructures : a key for energy efficiency at ultrascale level,” *Supercomput. Front. Innov.*, vol. 2, no. 2, pp. 105–131, 2015.

[5] A. Beloglazov, Y. C. Lee, and A. Zomaya, “A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems,” *Adv. Comput.*, vol. 82, pp. 47–111, 2011.

[6] Amazon_EC2, “Amazon EC2 Pricing,” 2018. [Online]. Available: https://aws.amazon.com/ec2/pricing/. [Accessed: 30-Oct-2018].

[7] Microsoft, “Microsoft Azure - Virtual Machines Pricing,” 2016. [Online]. Available: https://azure.microsoft.com/en-gb/pricing/details/virtual-machines/linux/. [Accessed: 30-Oct-2018].

[8] Google, “Google Cloud Pricing,” 2016. [Online]. Available: https://cloud.google.com/pricing/. [Accessed: 30-Oct-2018].

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

[10] M. Hinz, G. P. Koslovski, C. C. Miers, L. L. Pilla, and M. A. Pillon, “A Cost Model for IaaS Clouds Based on Virtual Machine Energy Consumption,” *J. Grid Comput.*, vol. 16, no. 3, pp. 493–512, Sep. 2018.

[11] D. Laganà, C. Mastroianni, M. Meo, and D. Renga, “Reducing the Operational Cost of Cloud Data Centers through Renewable Energy,” *Algorithms*, vol. 11, no. 10, p. 145, Sep. 2018.

[12] 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.

[13] H. T. Vu and S. Hwang, “A Traffic and Power-aware Algorithm for Virtual Machine Placement in Cloud Data Center,” *Int. J. Grid Distrib. Comput.*, vol. 7, no. 1, pp. 21–32, Feb. 2014.

[14] S. Dutta, S. Gera, A. Verma, and B. Viswanathan, “SmartScale: Automatic Application Scaling in Enterprise Clouds,” in *2012 IEEE Fifth International Conference on Cloud Computing*, 2012, pp. 221–228.

[15] M. Ficco, C. Esposito, F. Palmieri, and A. Castiglione, “A coral-reefs and Game Theory-based approach for optimizing elastic cloud resource allocation,” *Futur. Gener. Comput. Syst.*, vol. 78, pp. 343–352, Jan. 2018.

[16] H. Zhou, Q. Li, K.-K. R. Choo, and H. Zhu, “DADTA: A novel adaptive strategy for energy and performance efficient virtual machine consolidation,” *J. Parallel Distrib. Comput.*, vol. 121, pp. 15–26, Nov. 2018.

[17] M. Mao and M. Humphrey, “Auto-scaling to minimize cost and meet application deadlines in cloud workflows,” in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis on - SC ’11*, 2011, pp. 1–12.

[18] J. Yang *et al.*, “A cost-aware auto-scaling approach using the workload prediction in service clouds,” *Inf. Syst. Front.*, vol. 16, no. 1, pp. 7–18, Mar. 2014.

[19] Y. Jiang, C. Perng, T. Li, and R. Chang, “ASAP: A Self-Adaptive Prediction System for Instant Cloud Resource Demand Provisioning,” in *2011 IEEE 11th International Conference on Data Mining*, 2011, pp. 1104–1109.

[20] Q. Zhang, H. Chen, and Z. Yin, “PRMRAP: A Proactive Virtual Resource Management Framework in Cloud,” in *2017 IEEE International Conference on Edge Computing (EDGE)*, 2017, pp. 120–127.

[21] A. Gandhi, P. Dube, A. Karve, A. Kochut, and L. Zhang, “Modeling the Impact of Workload on Cloud Resource Scaling,” in *2014 IEEE 26th International Symposium on Computer Architecture and High Performance Computing*, 2014, pp. 310–317.

[22] Amazon_EC2, “Amazon EC2 Service Level Agreement,” 2013. [Online]. Available: https://aws.amazon.com/ec2/sla/. [Accessed: 01-Oct-2017].

[23] P. Berndt and A. Maier, “Towards Sustainable IaaS Pricing,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8193 LNCS, 2013, pp. 173–184.

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

[25] 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, 2013.

[26] A. I. Arutyun *et al.*, “Open Cirrus: A Global Cloud Computing Testbed,” *Computer (Long. Beach. Calif)*, vol. 43, no. 4, pp. 35–43, 2010.

[27] 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.

[28] S. Yeo and H. H. S. Lee, “Using mathematical modeling in provisioning a heterogeneous cloud computing environment,” *Computer*, vol. 44, no. 8, pp. 55–62, 2011.

[29] D. Armstrong, “Enhancing quality of service in cloud computing through novel resource management,” PhD Thesis, University of Leeds, 2013.

[30] 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.

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

[32] M. Al-Roomi, S. Al-Ebrahim, S. Buqrais, and I. Ahmad, “Cloud Computing Pricing Models: A Survey,” *Int. J. Grid Distrib. Comput.*, vol. 6, no. 5, pp. 93–106, Oct. 2013.

[33] P. M. Mell and T. Grance, “The NIST definition of cloud computing,” Gaithersburg, MD, 2011.

[34] H. Yang and M. Tate, “A Descriptive Literature Review and Classification of Cloud Computing Research,” *Commun. Assoc. Inf. Syst.*, vol. 31, no. 2, 2012.

[35] F. Liu *et al.*, “NIST cloud computing reference architecture,” IEEE, Gaithersburg, MD, 2011.

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

[37] 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, May 2010.

[38] KVM, “Kernel-based Virtual Machine.” [Online]. Available: https://www.linux-kvm.org/. [Accessed: 03-Apr-2018].

[39] Xen, “Xen Project.” [Online]. Available: https://xenproject.org/. [Accessed: 01-Mar-2019].

[40] VMware, “VMware Cloud.” [Online]. Available: https://www.vmware.com/. [Accessed: 01-Mar-2019].

[41] G. Li and M. Wei, “Everything-as-a-service platform for on-demand virtual enterprises,” *Inf. Syst. Front.*, vol. 16, no. 3, pp. 435–452, Jul. 2014.

[42] Google, “Google App Engine — Google Developers.” [Online]. Available: https://cloud.google.com/appengine/docs/?hl=en. [Accessed: 01-Mar-2019].

[43] Microsoft, “Windows Azure: Microsoft’s Cloud Platform | Cloud Hosting | Cloud Services.” [Online]. Available: https://azure.microsoft.com/en-us/. [Accessed: 01-Mar-2019].

[44] Rackspace, “Managed Cloud Services.” [Online]. Available: https://www.rackspace.com/en-gb/cloud. [Accessed: 01-Mar-2019].

[45] Cloudpedia, “Types of Cloud Deployment Models.” [Online]. Available: https://sites.google.com/site/cloudwikipedia/home/types-of-services/deployment-models-in-cloud-computing. [Accessed: 01-Mar-2019].

[46] S. Nanda and T.-C. Chiueh, “A Survey on Virtualization Technologies,” 2005.

[47] K. Hwang, J. Dongarra, and G. C. Fox, *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*, 1st ed. Morgan Kaufmann, 2011.

[48] Y. Li, Y. Wang, B. Yin, and L. Guan, “An Online Power Metering Model for Cloud Environment,” in *2012 IEEE 11th International Symposium on Network Computing and Applications*, 2012, pp. 175–180.

[49] OpenNebula, “The Simplest Cloud Management Experience.” [Online]. Available: https://opennebula.org/. [Accessed: 03-Apr-2018].

[50] OpenStack, “Open Source Software for Creating Private and Public Clouds.” [Online]. Available: https://www.openstack.org/. [Accessed: 01-Mar-2019].

[51] CloudStack, “Apache CloudStackTM - Open Source Cloud ComputingTM.” [Online]. Available: http://cloudstack.apache.org/. [Accessed: 01-Mar-2019].

[52] 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.

[53] OpenQRM, “Professional Open-Source Data Center and Cloud Management.” [Online]. Available: https://openqrm-enterprise.com/. [Accessed: 01-Mar-2019].

[54] EUCALYPTUS, “Eucalyptus.” [Online]. Available: https://www.eucalyptus.cloud/. [Accessed: 01-Mar-2019].

[55] Nimbus, “Nimbus is cloud computing for science.” [Online]. Available: http://www.nimbusproject.org/. [Accessed: 01-Mar-2019].

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

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

[58] Microsoft, “Microsoft Hyper-V.” [Online]. Available: https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/. [Accessed: 01-Mar-2019].

[59] Oracle, “Virtual Box.” [Online]. Available: http://www.virtualbox.org/. [Accessed: 01-Mar-2019].

[60] T. Kamarainen, Y. Shan, M. Siekkinen, and A. Yla-Jaaski, “Virtual machines vs. containers in cloud gaming systems,” in *2015 International Workshop on Network and Systems Support for Games (NetGames)*, 2015, vol. 2016-Janua, pp. 1–6.

[61] Cisco; Red Hat, “Linux Containers : Why They ’ re in Your Future and What Has to Happen First Application Delivery : Today ’ s Challenges,” 2014.

[62] Docker, “Enterprise Container Platform for High-Velocity Innovation.” [Online]. Available: https://www.docker.com/. [Accessed: 01-Mar-2019].

[63] B. Business, “How to Jump From Cloud to Cloud,” 2016. [Online]. Available: https://www.bloomberg.com/news/articles/2015-08-27/switching-cloud-providers-standards-sought-for-container-software. [Accessed: 01-Mar-2019].

[64] Linux, “LXC - Linux Containers.” [Online]. Available: https://linuxcontainers.org/. [Accessed: 01-Mar-2019].

[65] Cloud_Foundry, “Cloudfoundry warden manages isolated, ephemeral, and resource con- trolled environments.” [Online]. Available: https://github.com/cloudfoundry-attic/warden. [Accessed: 01-Mar-2019].

[66] J. Varia, “Architecting for the Cloud: Best Practices,” 2010.

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

[68] M. Ficco, M. Rak, S. Venticinque, L. Tasquier, and G. Aversano, “Cloud Evaluation: Benchmarking and Monitoring,” in *Quantitative Assessments of Distributed Systems*, no. April, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2015, pp. 175–199.

[69] SPEC, “Benchmark Overview — SPEC Cloud IaaS 2018 Benchmark.” [Online]. Available: https://www.spec.org/cloud_iaas2018/docs/faq.html. [Accessed: 12-Oct-2019].

[70] SPEC, “SPEC - Standard Performance Evaluation Corporation.” [Online]. Available: https://www.spec.org/. [Accessed: 12-Oct-2019].

[71] M. Forshaw, A. S. McGough, and N. Thomas, “HTC-Sim: a trace-driven simulation framework for energy consumption in high-throughput computing systems,” *Concurr. Comput. Pract. Exp.*, vol. 28, no. 12, pp. 3260–3290, Aug. 2016.

[72] POSIX, “POSIX.” [Online]. Available: http://www.opengroup.org/austin/papers/posix_faq.html. [Accessed: 12-Oct-2019].

[73] Stress-ng, “stress tests.” [Online]. Available: http://kernel.ubuntu.com/~cking/stress-ng/. [Accessed: 03-Apr-2018].

[74] J. Fabra, J. Ezpeleta, and P. Álvarez, “Reducing the price of resource provisioning using EC2 spot instances with prediction models,” *Futur. Gener. Comput. Syst.*, vol. 96, pp. 348–367, Jul. 2019.

[75] H. Wang, Q. Jing, R. Chen, and B. He, “Distributed systems meet economics: pricing in the cloud,” in *Proceeding HotCloud’10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, 2010, pp. 1–6.

[76] Jelastic, “Jelastic Pricing.” [Online]. Available: https://jelastic.com/pricing/. [Accessed: 01-Mar-2019].

[77] Amazon_EC2, “Amazon EC2 Spot Instances.” [Online]. Available: https://aws.amazon.com/ec2/spot/. [Accessed: 01-Mar-2019].

[78] Amazon_AWS, “Spot Instance Pricing History.” [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances-history.html. [Accessed: 01-Mar-2019].

[79] M. Aldossary, I. Alzamil, and K. Djemame, “Towards Virtual Machine Energy-Aware Cost Prediction in Clouds,” in *Proceedings of the 14h International Conference on Economics of Grids, Clouds, Systems, and Services (GECON’2017), Biarritz, France, September 19-21*, B. J. Pham C., Altmann J., Ed. Biarritz, France: Springer, Cham, 2017, pp. 119–131.

[80] M. Aldossary and K. Djemame, “Energy-based Cost Model of Virtual Machines in a Cloud Environment,” in *2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT)*, 2018, pp. 1–8.

[81] H. Jin, X. Wang, S. Wu, S. Di, and X. Shi, “Towards Optimized Fine-Grained Pricing of IaaS Cloud Platform,” *IEEE Trans. Cloud Comput.*, vol. 3, no. 4, pp. 436–448, Oct. 2015.

[82] S. Mireslami, L. Rakai, M. Wang, and B. H. Far, “Dynamic Cloud Resource Allocation Considering Demand Uncertainty,” *IEEE Trans. Cloud Comput.*, vol. PP, no. c, pp. 1–1, 2019.

[83] S. Ilager, K. Ramamohanarao, and R. Buyya, “ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation,” *Concurr. Comput. Pract. Exp.*, vol. 64, no. 2, p. e5221, Apr. 2019.

[84] X. Zhou, K. Li, C. Liu, and K. Li, “An Experience-Based Scheme for Energy-SLA Balance in Cloud Data Centers,” *IEEE Access*, vol. 7, pp. 23500–23513, 2019.

[85] G. Cook, T. Dowdall, D. Pomerantz, and Y. Wang, “Clicking Clean : How Companies are Creating the Green Internet,” 2014.

[86] Q. Ding, B. Tang, P. Manden, and J. Ren, “A learning-based cost management system for cloud computing,” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, vol. 2018-Janua, pp. 362–367.

[87] O. Belli, C. Loomis, and N. Abdennadher, “Towards a Cost-Optimized Cloud Application Placement Tool,” in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2016, pp. 43–50.

[88] M. S. Aslanpour, M. Ghobaei-Arani, and A. Nadjaran Toosi, “Auto-scaling web applications in clouds: A cost-aware approach,” *J. Netw. Comput. Appl.*, vol. 95, no. July, pp. 26–41, Oct. 2017.

[89] R. Chard, K. Chard, K. Bubendorfer, L. Lacinski, R. Madduri, and I. Foster, “Cost-Aware Elastic Cloud Provisioning for Scientific Workloads,” in *2015 IEEE 8th International Conference on Cloud Computing*, 2015, pp. 971–974.

[90] K. Djemame *et al.*, “Energy Efficiency Embedded Service Lifecycle : Towards an Energy Efficient Cloud Computing Architecture,” in *the Proceedings of the Workshop on Energy Efficient Systems (EES’2014) at ICT4S*, 2014, pp. 1–6.

[91] K. Djemame *et al.*, “PaaS-IaaS Inter-Layer Adaptation in an Energy-Aware Cloud Environment,” *IEEE Trans. Sustain. Comput.*, vol. 2, no. 2, pp. 127–139, Apr. 2017.

[92] I. Alzamil and K. Djemame, “Energy Prediction for Cloud Workload Patterns,” in *Proceedings of the 13th International Conference on Economics of Grids, Clouds, Systems and Services (GECON’2016), Athens, Greece, September 20-22*, 2017, pp. 160–174.

[93] A. Yousefipour, A. M. Rahmani, and M. Jahanshahi, “Energy and cost-aware virtual machine consolidation in cloud computing,” *Softw. Pract. Exp.*, vol. 48, no. 10, pp. 1758–1774, Apr. 2018.

[94] M. Aldossary, K. Djemame, I. Alzamil, A. Kostopoulos, A. Dimakis, and E. Agiatzidou, “Energy-aware cost prediction and pricing of virtual machines in cloud computing environments,” *Futur. Gener. Comput. Syst.*, vol. 93, pp. 442–459, Apr. 2019.

[95] M. Kumar and S. C. Sharma, “PSO-COGENT: Cost and energy efficient scheduling in cloud environment with deadline constraint,” *Sustain. Comput. Informatics Syst.*, vol. 19, no. January, pp. 147–164, Sep. 2018.

[96] S. S. Wagle, M. Guzek, and P. Bouvry, “Service Performance Pattern Analysis and Prediction of Commercially Available Cloud Providers,” in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2016, pp. 26–34.

[97] Z. Gong, X. Gu, and J. Wilkes, “PRESS: PRedictive Elastic reSource Scaling for cloud systems,” in *Proceedings of the 2010 International Conference on Network and Service Management, CNSM 2010*, 2010, vol. 2010, no. Cnsm, pp. 9–16.

[98] OW2, “RUBiS.” [Online]. Available: http://rubis.ow2.org/. [Accessed: 01-Mar-2019].

[99] Q. Huang, K. Shuang, P. Xu, J. Li, X. Liu, and S. Su, “Prediction-based Dynamic Resource Scheduling for Virtualized Cloud Systems,” *J. Networks*, vol. 9, no. 2, pp. 375–383, Feb. 2014.

[100] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N. T. Hieu, and H. Tenhunen, “Energy-Aware VM Consolidation in Cloud Data Centers Using Utilization Prediction Model,” *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 524–536, Apr. 2019.

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

[102] J. Yang, C. Liu, Y. Shang, Z. Mao, and J. Chen, “Workload Predicting-Based Automatic Scaling in Service Clouds,” in *2013 IEEE Sixth International Conference on Cloud Computing*, 2013, pp. 810–815.

[103] W. L. Bircher and L. K. John, “Complete System Power Estimation Using Processor Performance Events,” *IEEE Trans. Comput.*, vol. 61, no. 4, pp. 563–577, Apr. 2012.

[104] J. C. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, and R. K. Gupta, “Evaluating the Effectiveness of Model-based Power Characterization,” in *The 2011 USENIX Conference on USENIX Annual Technical Conference*, 2011, pp. 1–14.

[105] J. W. Smith, A. Khajeh-Hosseini, J. S. Ward, and I. Sommerville, “CloudMonitor: Profiling Power Usage,” in *2012 IEEE Fifth International*

Conference on Cloud Computing, 2012, pp. 947–948.

[106] J. von Kistowski, M. Deffner, and S. Kounev, “Run-Time Prediction of Power Consumption for Component Deployments,” in *2018 IEEE International Conference on Autonomic Computing (ICAC)*, 2018, pp. 151–156.

[107] A. T. Makaratzis, K. M. Giannoutakis, and D. Tzovaras, “Energy modeling in cloud simulation frameworks,” *Futur. Gener. Comput. Syst.*, vol. 79, no. 2018, pp. 715–725, Feb. 2018.

[108] 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.

[109] J. Subirats and J. Guitart, “Assessing and forecasting energy efficiency on Cloud computing platforms,” *Futur. Gener. Comput. Syst.*, vol. 45, pp. 70–94, Apr. 2015.

[110] N. Roy, A. Dubey, and A. Gokhale, “Efficient autoscaling in the cloud using predictive models for workload forecasting,” in *Proceedings - 2011 IEEE 4th International Conference on Cloud Computing, CLOUD 2011*, 2011, pp. 500–507.

[111] U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, “Kingfisher: A system for elastic cost-aware provisioning in the cloud,” *Tech. Rep. UM-CS-2010-005*, 2010.

[112] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, “Performance and energy modeling for live migration of virtual machines,” *Cluster Comput.*, vol. 16, no. 2, pp. 249–264, Jun. 2013.

[113] F. Farahnakian, R. Bahsoon, P. Liljeberg, and T. Pahikkala, “Self-Adaptive Resource Management System in IaaS Clouds,” in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, 2016, pp. 553–560.

[114] O. Alrajeh, M. Forshaw, A. S. McGough, and N. Thomas, “Simulation of Virtual Machine Live Migration in High Throughput Computing Environments,” in *2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2018, pp. 1–8.

[115] A. Beloglazov and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers,” *Concurr. Comput. Pract. Exp.*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.

[116] O. Alrajeh, M. Forshaw, and N. Thomas, “Machine Learning Models for Predicting Timely Virtual Machine Live Migration,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10497 LNCS, A. Irfan and C. Andrea, Eds. 2017, pp. 169–183.

[117] K. Ye, X. Jiang, D. Huang, J. Chen, and B. Wang, “Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments,” in *2011 IEEE 4th International Conference on Cloud*

Computing, 2011, pp. 267–274.

[118] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, and H. Tenhunen, “Utilization Prediction Aware VM Consolidation Approach for Green Cloud Computing,” in *2015 IEEE 8th International Conference on Cloud Computing*, 2015, pp. 381–388.

[119] M. R. Hines and K. Gopalan, “Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning,” in *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments - VEE ’09*, 2009, p. 51.

[120] B. R. Raghunath and B. Annappa, “Virtual Machine Migration Triggering using Application Workload Prediction,” *Procedia Comput. Sci.*, vol. 54, pp. 167–176, 2015.

[121] H. Zhao, Q. Zheng, W. Zhang, Y. Chen, and Y. Huang, “Virtual machine placement based on the VM performance models in cloud,” in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, 2015, no. 28, pp. 1–8.

[122] T. C. Ferreto, M. A. S. Netto, R. N. Calheiros, and C. A. F. De Rose, “Server consolidation with migration control for virtualized data centers,” *Futur. Gener. Comput. Syst.*, vol. 27, no. 8, pp. 1027–1034, Oct. 2011.

[123] A. Beloglazov and R. Buyya, “Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1366–1379, Jul. 2013.

[124] F. Xu, F. Liu, L. Liu, H. Jin, B. Li, and B. Li, “iAware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud,” *IEEE Trans. Comput.*, vol. 63, no. 12, pp. 3012–3025, Dec. 2014.

[125] A. Beloglazov and R. Buyya, “Energy Efficient Resource Management in Virtualized Cloud Data Centers,” in *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, 2010, pp. 826–831.

[126] 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, May 2012.

[127] A. Beloglazov and R. Buyya, “Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers,” in *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science - MGC ’10*, 2010, no. December 2010, pp. 1–6.

[128] M.-H. Malekloo, N. Kara, and M. El Barachi, “An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments,” *Sustain. Comput. Informatics Syst.*, vol. 17, pp. 9–24, Mar. 2018.

[129] A. Verma, P. Ahuja, and A. Neogi, “pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5346 LNCS, 2008, pp. 243–264.

[130] O. Alrajeh, M. Forshaw, and N. Thomas, “Performance of Virtual Machine Live Migration with Various Workloads,” in *32nd Annual UK Performance Engineering Workshop & Cyber Security Workshop*, 2016, pp. 28–39.

[131] M. Zakarya and L. Gillam, “An Energy Aware Cost Recovery Approach for Virtual Machine Migration,” in *Proceedings of the 13th International Conference on Economics of Grids, Clouds, Systems and Services (GECON’2016), Athens, Greece, September 20-22*, Athens, Greece: Springer, 2017, pp. 175–190.

[132] Amazon Web Services, “AWS.” [Online]. Available: https://aws.amazon.com/. [Accessed: 01-Mar-2019].

[133] A. Y. Nikravesh, S. A. Ajila, and C.-H. Lung, “An autonomic prediction suite for cloud resource provisioning,” *J. Cloud Comput.*, vol. 6, no. 1, p. 3, Dec. 2017.

[134] F. Lombardi, A. Muti, L. Aniello, R. Baldoni, S. Bonomi, and L. Querzoni, “PASCAL: An architecture for proactive auto-scaling of distributed services,” *Futur. Gener. Comput. Syst.*, vol. 98, pp. 342–361, Sep. 2019.

[135] M. Tighe and M. Bauer, “Integrating cloud application autoscaling with dynamic VM allocation,” in *2014 IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–9.

[136] M. Tighe and M. Bauer, “Topology and Application Aware Dynamic VM Management in the Cloud,” *J. Grid Comput.*, vol. 15, no. 2, pp. 273–294, Jun. 2017.

[137] W. Dawoud, I. Takouna, and C. Meinel, “Elastic VM for Cloud Resources Provisioning Optimization,” in *Communications in Computer and Information Science*, vol. 190 CCIS, no. PART 1, 2011, pp. 431–445.

[138] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, “Efficient resource provisioning in compute clouds via VM multiplexing,” in *Proceeding of the 7th international conference on Autonomic computing - ICAC ’10*, 2010, pp. 11–20.

[139] M. Aldossary and K. Djemame, “Performance and Energy-based Cost Prediction of Virtual Machines Live Migration in Clouds,” in *Proceedings of the 8th International Conference on Cloud Computing and Services Science*, 2018, pp. 384–391.

[140] M. Aldossary and K. Djemame, “Performance and Energy-Based Cost Prediction of Virtual Machines Auto-Scaling in Clouds,” in *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2018, pp. 502–509.

[141] I. Corporation, “Intel ® 64 and IA-32 Architectures Software Developer ’ s Manual Volume 3A : System Programming Guide , Part 1,” in *System*, vol. 3, no. 253666, 2010, pp. 1807–1814.

[142] Intel, “Intelligent Platform Management Interface (IPMI),” 2015. [Online]. Available: https://www.intel.com/content/www/us/en/servers/ipmi/ipmi-home.html. [Accessed: 17-Jul-2018].

[143] WattsUp, “Watts Up? Plug Load Meters.” [Online]. Available: https://www.wattsupmeters.com/secure/products.php?pn=0. [Accessed: 03-Apr-2018].

[144] I. A. M. Alzamil, “Energy-Aware Profiling and Prediction Modelling of Virtual Machines in Cloud Computing Environments,” PhD Thesis, University of Leeds, 2017.

[145] A. Kansal, F. Zhao, and A. A. Bhattacharya, “Virtual Machine Power Metering and Provisioning,” in *SoCC ’10 Proceedings of the 1st ACM symposium on Cloud computing*, 2010, pp. 39–50.

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

[147] P. Garraghan, I. S. Moreno, P. Townend, and J. I. E. 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.

[148] R. Kavanagh, D. Armstrong, K. Djemame, D. Sommacampagna, and L. Blasi, “Towards an Energy-Aware Cloud Architecture for Smart Grids,” in *Proceedings of the 12th International Conference on Economics of Grids, Clouds, Systems and Services (GECON’2015), Cluj-Napoca, Romania, September 15-17*, 2016, pp. 190–204.

[149] 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.

[150] Zabbix, “The Enterprise-Class Monitoring Solution for Everyone.” [Online]. Available: https://www.zabbix.com/. [Accessed: 03-Apr-2018].

[151] Rackspace, “Cloud Servers Pricing and Cloud Server Costs.” [Online]. Available: https://www.rackspace.com/cloud/servers/pricing. [Accessed: 03-Apr-2018].

[152] ElasticHosts, “Pricing - ElasticHosts Linux.” [Online]. Available: https://www.elastichosts.co.uk/pricing/. [Accessed: 03-Apr-2018].

[153] VMware, “OnDemand Pricing Calculator.” [Online]. Available: http://vcloud.vmware.com/uk/service-offering/pricing-calculator/on-demand. [Accessed: 03-Apr-2018].

[154] CompareMySolar, “Electricity Price Electricity Price per kWh Comparison of Big Six Energy Companies - CompareMySolar.co.uk.” [Online]. Available: http://blog.comparemysolar.co.uk/electricity-price-per-kwh-comparison-of-big-six-energy-companies/. [Accessed: 03-Apr-2018].

[155] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

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

[157] C. Liu, S. C. H. Hoi, P. Zhao, and J. Sun, “Online ARIMA Algorithms for Time Series Prediction,” in *The Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1867–1873.

[158] G. E. P. Box and D. R. Cox, “An Analysis of Transformations,” *J. R. Stat. Soc. Ser. B*, vol. 26, pp. 211–252, 1964.

[159] R, “The R Project for Statistical Computing.” [Online]. Available:

http://www.r-project.org/. [Accessed: 03-Apr-2019].

[160] 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.

[161] R. J. Hyndman and G. Athanasopoulos, “Forecasting: Principles and Practice,” *OTexts*, 2013. [Online]. Available: http://otexts.org/fpp/. [Accessed: 03-Apr-2018].

[162] IBM Group, “An Architectural Blueprint for Autonomic Computing, Technical Whitepaper (Fourth Edition),” 2006.

[163] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, “Cost of virtual machine live migration in clouds: A performance evaluation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5931 LNCS, pp. 254–265.

[164] W. Li, Y. Xia, M. Zhou, X. Sun, and Q. Zhu, “Fluctuation-Aware and Predictive Workflow Scheduling in Cost-Effective Infrastructure-as-a-Service Clouds,” *IEEE Access*, vol. 6, no. c, pp. 61488–61502, 2018.

[165] T. L. J. Miguel-alonso and J. A. Lozano, “A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments,” *J. Grid Comput.*, vol. 12, no. 4, pp. 559–592, 2014.