UNIVERSITY OF LEEDS

This is a repository copy of *Energy-aware cost prediction and pricing of virtual machines in cloud computing environments*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/138640/

Version: Accepted Version

Article:
Aldossary, M orcid.org/0000-0001-6772-700X, Djemame, K orcid.org/0000-0001-5811-5263, Alzamil, I et al. (3 more authors) (2019) Energy-aware cost prediction and pricing of virtual machines in cloud computing environments. Future Generation Computer Systems, 93. pp. 442-459. ISSN 0167-739X

https://doi.org/10.1016/j.future.2018.10.027

Crown Copyright © 2018 Published by Elsevier B.V. All rights reserved. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/.

Reuse
This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can't change the article in any way or use it commercially. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Energy-Aware Cost Prediction and Pricing of Virtual Machines in Cloud Computing Environments

Mohammad Aldossary[a,b], Karim Djemame[a], Ibrahim Alzamil[c], Alexandros Kostopoulos[d], Antonis Dimakis[d], Eleni Agiatzidou[d]

[a]*School of Computing, University of Leeds, UK*
[b]*Prince Sattam Bin Abdulaziz University, KSA*
[c]*Majmaah University, KSA*
[d]*Department of Informatics*
Athens University of Economics and Business, Greece

Abstract

With the increasing cost of electricity, Cloud providers consider energy consumption as one of the major cost factors to be maintained within their infrastructure. Consequently, various proactive and reactive management mechanisms are used to efficiently manage the cloud resources and reduce the energy consumption and cost. These mechanisms support energy-awareness at the level of Physical Machines (PM) as well as Virtual Machines (VM) to make corrective decisions. This paper introduces a novel Cloud system architecture that facilitates an energy aware and efficient cloud operation methodology and presents a cost prediction framework to estimate the total cost of VMs based on their resource usage and power consumption. The evaluation on a Cloud testbed show that the proposed energy-aware cost prediction framework is capable of predicting the workload, power consumption and estimating total cost of the VMs with good prediction accuracy for various Cloud application workload patterns. Furthermore, a set of energy-based pricing schemes are defined, intending to provide the necessary incentives to create an energy-efficient and economically sustainable ecosystem. Further evaluation results show that the adoption of energy-based pricing by cloud and application providers creates additional economic value to both under different market conditions.

Keywords: Cloud Computing, Energy Efficiency, Cost Estimation, Workload Prediction, Energy Prediction, Pricing Schemes

1. Introduction

The emergence of cloud computing as an IT service has seen the provision of computing power and storage away from companies and organisations. Cloud remote data centres, managed by cloud providers, handle the services required by customers, including Small and Medium Enterprises (SMEs) in a centralised and controlled environment rather a local IT system. These providers make use of virtualisation in the management of ICT resources, which provides a simplified server administration, improved resource utilisation, and reduced IT costs.

However, with the wide adoption of Cloud Computing, energy consumption has become one of the main issues for Cloud providers to address. A Cloud infrastructure along with its cooling resources consume a large amount of energy to operate, which may cause ecological and economic issues. From the economical perspective, Cloud providers consider energy consumption as one of the key cost factors with a substantial impact on the operational cost of the Cloud infrastructure [1]. Therefore, various energy efficient techniques have been introduced to help Cloud providers reduce the energy cost of their infrastructure, which can then lead to reducing the cost of operational expenditure (OPEX) and having less negative impact on the environment. Cost mechanisms offered by Cloud providers have become sophisticated, as customers are charged per month, hour or minute for the services they use. Nevertheless, there are still limited as customers are charged based on a predefined tariff for the resources usage which include CPU, memory, storage and network. This pre-defined tariff does not consider the variable cost of energy [2]. Consequently, modelling a new cost mechanism for services offered that can be adjusted to the actual energy costs has become an interesting research topic.

The impact of energy consumption is not only dependent on the efficiency of the physical resources, but also on the strategies deployed to manage these resources as well as the efficient design of the applications running on these resources [3]. Different methods have been used to efficiently manage cloud resources, all of which can be based on certain thresholds, called *reactive*, or based on prediction, called *proactive* [4]. For example, once an 80% CPU utilisation threshold is exceeded, a corrective action takes place such as adding more resources to avoid service performance degradation. Proactive methods have the advantage of taking corrective actions at an early stage to prevent Service Level Agreement (SLA) violation and maintain the expected service

performance. To efficiently design cloud applications, applications' designers and developers should be provided with energy-aware and cost information for supporting the task of optimising energy efficiency resulting from running services in cloud environments. As discussed in [5], having appropriate tools for energy monitoring is essential to support energy-awareness and contributes to energy optimisation in all layers of the Cloud stack. Furthermore, estimating the total cost of cloud services can help make effective strategies and energy efficient resource allocation methods [5]. Thus, managing the Cloud paradigm in all different levels and reducing the energy consumption has received a lot of attention in the literature as it can result in reduction of OPEX costs for the Cloud providers.

Another important aspect is to consider novel pricing schemes, intending to provide the necessary incentives to create an energy-efficient and economically sustainable ecosystem. Pricing in cloud computing has been studied extensively in the past and most approaches consist of a combination of a fixed or variable price per VM instance and an additional usage charge based on the actual use of computing resources, such as CPU cycles, network bandwidth, memory and storage. Some cloud providers employ even simpler pricing schemes, such as monthly or yearly subscriptions. However, none of the aforementioned schemes provide incentives for efficient energy consumption. One candidate solution could be the adoption of energy-aware pricing by the cloud service providers for achieving a more efficient resource usage.

Additionally, to evaluate the effect of pricing, one needs to consider the actions taken by all the economic agents involved. For example, a price increase by an Infrastructure as a Service (IaaS) provider does not necessarily lead to an increase in its profits, as the demand of applications for VMs might drop considerably. For this reason, a microeconomic model is considered, which incorporates the actions of IaaS as well as Platform as a Service (PaaS) providers, applications and their users. Since an action of any of these agents triggers a chain of subsequent responses by the others, determining the equilibrium of such interactions is an interesting problem.

The aim of this research is to enable energy-awareness of resource usage at virtual level in cloud computing environments, which contributes to overcome the challenge of identifying energy usage for the VMs. Also, this research aims to predict the workload, energy consumption and estimate total cost of the VMs based on specific cloud workload patterns. The outcome of this research can be used to help make efficient decisions supported with

energy-awareness and cost estimation. This paper's main contributions are summarised as follows:

- a Cloud system architecture that includes the required components to support energy-awareness and total cost estimation of Cloud infrastructure services;
- an energy-aware model that fairly attributes the energy consumption to heterogeneous and homogeneous VMs in Clouds;
- an energy-aware framework for predicting the usage and cost of heterogeneous and homogeneous VMs by considering their resource and power consumption;
- an adoption analysis of the proposed energy-based pricing schemes by cloud and application providers.

This paper is organised as follows: Section 2 introduces the proposed cloud system architecture. The energy-aware VM model and the energy-aware cost prediction framework are presented in Sections 3 and 4, respectively. In Section 5, a set of innovative energy-based pricing schemes are proposed. Section 6 presents the experimental set up and design. Section 7 includes the evaluation of the proposed cost estimation framework, as well as the economic implications of energy-aware pricing under different market conditions. Section 8 discusses the related work, and Section 9 concludes this research and discusses future steps.

2. System Architecture

In this Section, an architecture that supports energy-awareness in different levels of the Cloud stack while at the same time aware of the impact on other quality characteristics of the overall cloud system such as performance and cost is proposed. Figures 1-3 provide an overview of the proposed architecture [6]. It includes the high-level interactions of all components, is separated into three distinct layers whose interaction supports the standard Cloud service model: construct, deploy and operate/re-configure. Next, details on the interactions of the architectural components are discussed.

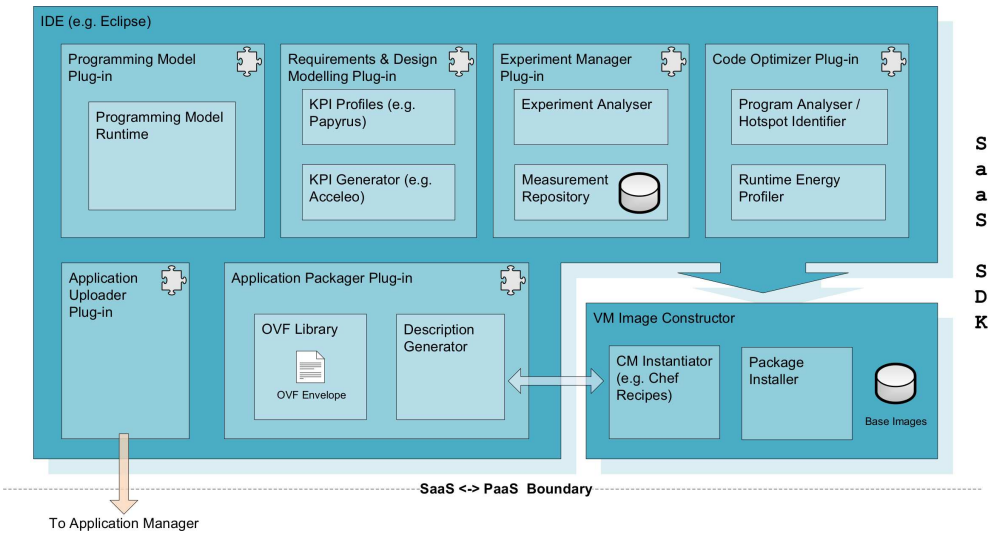

Figure 1: SaaS Architecture - Application Construction Kit.

2.1. Layer 1 - SaaS

In the SaaS layer a set of components interact to facilitate the modelling, design and construction of a Cloud application. The components aid in evaluating energy consumption of a Cloud application during its construction. A number of plug-ins are provided for a frontend Integrated Development Environment (IDE) as a means for developers to interact with components within this layer. A number of packaging components are also made available to enable provider agnostic deployment of the constructed cloud application, while also maintaining energy awareness.

The IDE is intended to be the main entry point to the infrastructure for service designers and developers. The idea is that the IDE integrates the graphical interfaces to the different tools available in the SaaS layer, thus offering a unified and integrated view to users. The *Programming Model Plug-in* (PM plug-in) provides a graphical interface to use the Programming Model and supporting tools to enable the development, analysis and profiling of an application in order to improve energy efficiency. On the other hand, the *Programming Model* provides the service developers with a way to implement services composed of source code, legacy applications executions and external Web services [7]. Although these complex services are written in a sequential fashion without APIs, the applications are instrumented so they call the Programming Model Runtime to be executed in parallel.

The *Requirement and Design Modelling Plugins* are initially used during the system testing phase of a SaaS application. In cases of iterative or incremental development, this means that these SaaS Modelling tools can be used at the end of each iteration whose results provides an executable part of the SaaS application.

The *Experiment Manager (EM Plug-in)* is used prior to a SaaS application deployment. It assumes that a current SaaS application version has an executable version on which integration and system tests can be performed. The DEM helps a SaaS development team in cooperation with a SaaS provider to determine what deployment configuration alternatives of their SaaS application is likely to provide the most effectiveness business operation. In particular, the DEM will assist in managing experiments where application representative workloads are exercised on different deployment configuration alternatives of a SaaS application version to obtain measurements on cost, energy behaviour and time performance behaviour of each workload.

The *Code Optimizer* plays an essential role in the reduction of energy consumed by an application. This is achieved through the adaptation of the software development process and by providing SaaS software developers the ability to directly understand the energy foot print of the code they write.

Other components in this layer include 1) the *Application Packager* component is in charge of packaging applications. This component takes into account input from the Requirements and Design Modelling Plug-in in the Open Virtualisation Format (OVF) to package the software with the different requirements. It also generates a Service Manifest to submit to the VM Image Constructor; 2) the *VM Image Constructor* (VMIC) uses the application packages and the service manifest or application descriptor to create VM images that can be deployed in the PaaS layer, and 3) the *Application Uploader* interacts with the PaaS Application Manager to register the final VMs ready for deployment.

2.2. Layer 2 - PaaS

The PaaS layer provides middleware functionality for a Cloud application and facilitates the deployment and operation of the application as a whole. Components within this layer are responsible for selecting the most energy appropriate provider for a given set of energy requirements and tailoring the application to the selected providers hardware environment. The *Application Manager* (AM) component manages the user applications that are described as virtual appliances, formed by a set of VMs that are interconnected between

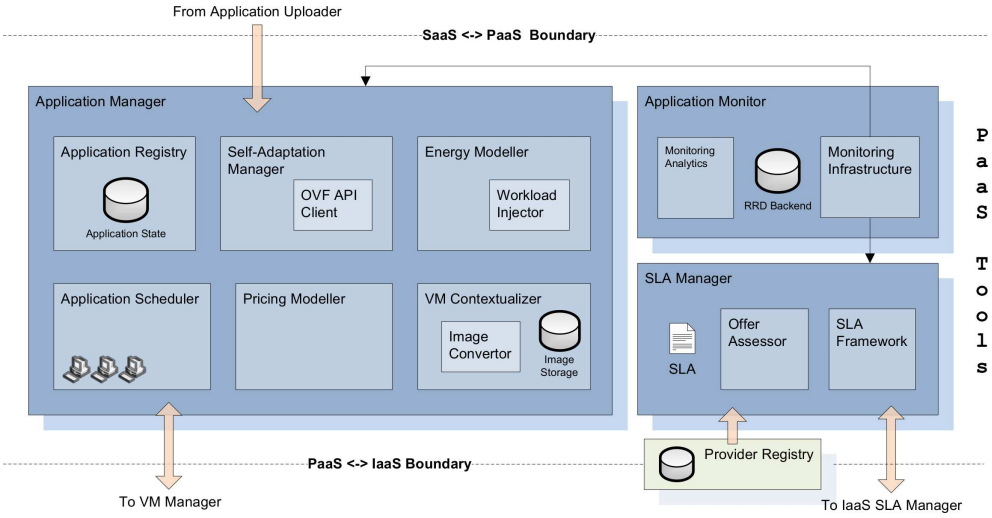

Figure 2: PaaS Architecture - Application Deployment.

them. The *Energy Modeller* aims to gather and manage energy related information throughout the whole Cloud Service lifecycle and Cloud layers: from requirement level Key Performance Indicators (KPIs) to programming model annotations down to PaaS and IaaS level measurements made through the monitoring agents present at those levels. The energy modeller provides an interface to estimate the energy cost of a PaaSs KPIs, and the provided estimations assist in the selection of the appropriate IaaS provider for running the application. Moreover, it provides aggregated measurements of energy consumption (Wh) and average instant power (W) per each application and its events as required by other components such as the *Pricing Modeller*, which needs to know the current energy consumption to get billing information, but also forecast the price change of an application deployment/re-deployment. It also provides energy-aware cost estimation related to the operation of applications on top of VMs on a specific IaaS provider. The role of the *Virtual Machine Contextualizer* (VMC) is to embed software dependencies of a service into a VM image and configure these dependencies at runtime via an infrastructure agnostic contextualization mechanism. Additionally, the VMC enables the use of energy probes for the gathering of VM level energy performance metrics. *Application level monitoring* is also accommodated for here, in addition to support for *Service Level Agreement* (SLA) negotiation. The *Self-Adaptation manager (PaaS SAM)* is the principle component in this

layer for deciding on the adaptation required to maintain SLAs. Its overall aim is to manage the trade-offs between energy, performance and cost during adaptation at runtime. The PaaS SAM is notified of the need to perform an adaptation by the SLA manager.

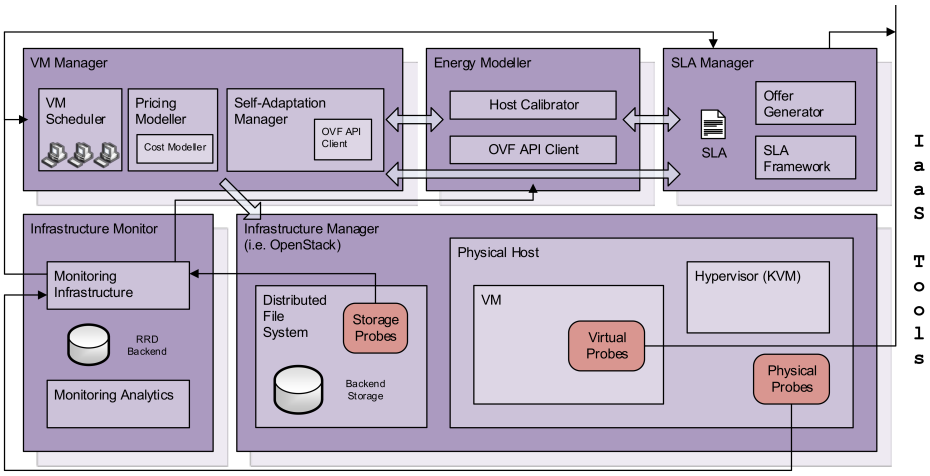

Figure 3: IaaS Architecture - Application Operation and Re-Configuration.

2.3. Layer 3 - IaaS

In the IaaS layer the admission, allocation and management of virtual resource are performed through the orchestration of a number of components. Energy consumption is monitored, estimated and optimised using translated PaaS level metrics. These metrics are gathered via a monitoring infrastructure and a number of software probes.

The *Virtual Machine Manager* (VMM) component is responsible for managing the complete life cycle of the virtual machines that are deployed in a specific infrastructure provider. The goal of the *Energy Modeller* is to gather and manage energy related information throughout the whole Cloud Service lifecycle and Cloud layers. This components core responsibility is to provide energy usage estimates by presenting the relevant KPIs for a virtual machine deployment on the infrastructure provided, see Section 3. This will include cost trade off analysis based on sources such as prior experience, the application profile as defined in the SLA, which is subsequently translated into infrastructure level KPIs, and finally from current up to date monitoring information from the deployment environment. The *SLA Manager* is

responsible for managing SLA negotiation requests at IaaS level. It interacts with the VM Manager to get the status of the available resources in order to determine the SLA offer and the *Pricing Modeller* to assign a price to the offered terms. The goal of the Pricing Modeller is to provide energy-aware cost estimation related to the operation of the physical resources managed by the IaaS provider and used by specific VMs, see Section 4. The *Infrastructure Manager* (IM) manages the physical infrastructure and redirects requests to hardware components. It maintains lists of hardware energy-meters, physical cluster nodes, network components and storage devices. External components can obtain and manipulate the state of the infrastructure through a common API that is independent of the actual hardware. The IM provides power consumption information for each cluster node. Furthermore, it IM requires an authentication for all operations which ensures protection against attacks as well as a sufficient separation of different parties. Finally, the ability to self-adapt at operation time which is supported by the *Self-Adaptation Manager (IaaS SAM)* is needed to keep the cloud infrastructure in an optimal state during its operation.

2.4. Layers Interaction

The focus on performance, cost and energy can be seen in each layer, with each component adding to the ability of the overall architecture to adapt [4]. The SaaS components not only support the energy efficiency goal but provide means of packaging Cloud applications in a way that enables provider agnostic deployment thanks to the interaction with the PaaS Application Manager through the *Application Uploader*.

The captured application requirements are realised in the PaaS layer by the application manager, which enables the deployment of the application on a cloud infrastructure thanks to the *Application Scheduler*. Self-Adaptation then continues in the PaaS layer through the collaboration between the Application manager and other key components, e.g. the Self-Adaptation manager. The SLA manager continually monitors SLA conformance with the aid of the Application Monitor while the Self-Adaptation Manager makes the decisions of when to adapt the application through horizontal scaling.

In the IaaS layer, the VMM is at the heart of the adaptation at this layer. Unlike the PaaS Layer that focusses on application level metrics the VMM focuses on optimising the VMs both at deployment and again at runtime. In order to do this it utilises energy and pricing modellers as well as key performance data from the infrastructure monitor and performs rescheduling

in order to adapt either on particular events such as submission of new VMs or periodically.

3. Energy-Aware Virtual Machine Model

The power consumption of a PM can be directly measured and mainly consists of two parts, the idle and active power. The idle power consumption is consumed when the PM is turned on but not running any workload, and the active power consumption is the induced power to the PM when it is running some workload. Thus, the total power consumption of the PM is equal to its idle power consumption plus its active power consumption.

As the case with the PM, the total power consumption of a VM equals its idle power consumption plus its active power consumption. However, the power consumption of VMs is difficult to identify and not directly measured. Hence, the power consumption of VMs can be inferred from their underlying PMs, which is still difficult to achieve.

A PM can run one or many VMs at the same time, and these VMs can be homogeneous or heterogeneous based on their characteristics, e.g. the number of Virtual CPUs (vCPUs) for each VM. Thus, these conditions should be taken into consideration when modelling and identifying the power consumption for the VMs.

Different energy models and mechanisms have been introduced in previous work to identify the energy consumption of VMs based on the energy consumption of their underlying PMs. Some of these models, as presented in [8], only attribute the PMs active energy to the VMs. Other models, as presented in [9] attribute both of the PMs idle and active energy to the VMs. Nevertheless, these introduced models do not consider a fair attribution the PMs idle and active energy to homogeneous and heterogeneous VMs running concurrently.

Thus, a new energy-aware model is introduced to overcome the above limitations of the existing VM energy models. This new energy model attributes the PMs idle and active energy consumption fairly to homogeneous and heterogeneous VMs running on the same PM.

Many of the existing approaches model and identify the energy consumption in PMs, as in [10, 11], and the energy consumption in VMs, as in [12, 9], by considering only the CPU utilisation. Understanding how the resource usage affects the power consumption is required. Further, an experimental study was carried out to investigate the effect of the resource usage (CPU,

RAM, disk and network) on the power consumption. The findings [13, 14, 15] show that the CPU utilisation correlates well with the power consumption, which is supported in other work, for example [11, 16, 17]. Thus, the work introduced in this paper follows the same approach and takes into account the CPU utilisation only when modelling and identifying the energy consumption for the VMs.

The energy-aware model introduced in this paper works by fairly attributing the PMs idle energy to VMs based on the number of vCPUs assigned to each VM. As shown in Equation 1, $PMx_{IdlePwr}$ is the idle power consumption of the PM where the VMs are hosted; $VMx_{ReqvCPUs}$ is the number of the vCPUs assigned to the given VMx; $VMcount$ is the number of VMs running on the same PM; and $VMy_{ReqvCPUs}$ is the number of vCPUs assigned to a member of the VMs set hosted by the same PM. In this way, the idle energy of the PM is fairly attributed to homogeneous and heterogeneous VMs by considering the size of each VM in terms of the vCPUs assigned to them.

$$VMx_{IdlePwr} = PMx_{IdlePwr} \times \left(\frac{VMx_{ReqvCPUs}}{\sum_{y=1}^{VMcount} VMy_{ReqvCPUs}}\right) \tag{1}$$

Further, the PMs active energy is fairly attributed to the VMs based on the VM CPU utilisation as well as the number of vCPUs assigned to each VM. As shown in Equation 2, PMx_{Pwr} is the total power consumption of the PM, from which the PMs idle power is deducted in order to identify the PMs active power; VMx_{Util} is the CPU utilisation of the given VMx; and VMy_{Util} is the CPU utilisation of a member of the VMs set hosted by the same PM. This way, the active energy of the PM is fairly attributed to heterogeneous and homogeneous VMs by considering the VM CPU utilisation and number of vCPUs assigned for each VM.

$$\begin{aligned} VMx_{ActivePwr} = (PMx_{Pwr} - PMx_{IdlePwr}) \times \\ \left(\frac{VMx_{(Util \times ReqvCPUs)}}{\sum_{y=1}^{VMcount} VMy_{(Util \times ReqvCPUs)}}\right) \end{aligned} \tag{2}$$

Therefore, the total power consumption for each VM at any given time can be identified by summing up its both idle and active power consumption, as shown in Equation 3.

$$VMx_{Pwr} = VMx_{IdlePwr} + VMx_{ActivePwr} \tag{3}$$

Hence, the presented energy-aware model can fairly attribute the idle and active energy consumption of a PM to the same or different sizes of VMs in terms of the allocated vCPUs for each VM. For instance, when both a small VM with 1 vCPU and a large VM with 3 vCPUs are being fully utilised on the same PM, the large VM would have triple the value in terms of energy consumption as compared to the small VM. This way the energy consumption can be fairly attributed based on the actual physical CPU resources used by each VM. Further, the presented model has revealed that a large portion of the VMs total energy represents idle energy, which is attributed to the underlying PM idle energy. 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.

4. Energy-Aware Cost Prediction Framework

Cost mechanisms offered by Cloud service providers have become even more sophisticated, as customers are charged per month, hour or minute based on the resources they utilised. Nevertheless, there are still limited as customers are charged based on a pre-defined tariff for the resource usage. This pre-defined tariff does not consider the variable cost of energy [2]. Measuring or predicting the current power consumption is difficult and cannot be performed directly at the VM level. 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-aware cost prediction framework that aims to predict the workload and power consumption as well as estimate the total cost of the VMs during service operation is introduced. The VMs workload (CPU, memory, disk and network) is firstly predicted. Then, the predicted VM CPU utilisation is correlated to PM workload characterised by (CPU utilisation) in order to estimate the PM power consumption, from which the predicted VM 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, this framework 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 [15].

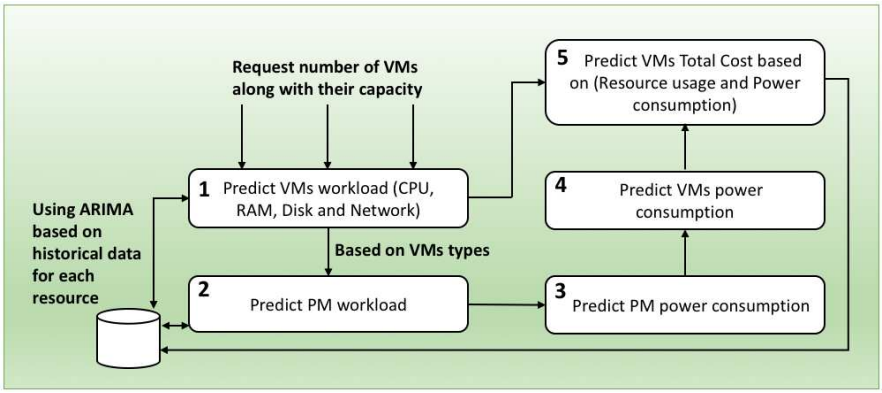

Figure 4: Energy-Aware Cost Prediction Framework.

4.1. VM 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 Autoregressive Integrated Moving Average (ARIMA) model, the VM workload is then predicted based on historical workload patterns retrieved from a knowledge database. There are five different types of workload patterns that can be experienced in cloud applications [18]; and two types of these workload patterns, namely static and periodic, are considered for the historical data to be used in this framework. A static workload pattern occurs when an application is experiencing the same and stable resource utilisation over a period of time. A periodic workload pattern can occur when an application is experiencing repeated resource utilisation peaks in time intervals [18].

The ARIMA model is a time series prediction model that has been used widely in different domains owing to its sophistication and accuracy [19]. A number of work, as in [20] have used ARIMA model to predict workload in the cloud computing domain; though their objectives do not consider predicting the energy consumption. Hence, the same approach using ARIMA model is applied in this work to predict the workload, but with the objectives toward predicting the energy consumption and estimating the total cost of VMs. Unlike other prediction methods, like sample average, ARIMA takes multiple inputs as historical observations and outputs multiple future observations depicting the seasonal trend. It can be used for seasonal or non-seasonal

time-series data. The type of seasonal ARIMA model is used in this work 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 [21] and data differencing methods are used to make these data stationary. Further, the model selection of ARIMA can be automatically processed in R package [22] using the ***auto.arima*** function, which selects the best fit model of ARIMA based on Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) value [19].

After predicting the VM workload using the ARIMA model based on historical data, the next steps take place to predict the PM workload and the PM/VM power consumption using regression models.

4.2. PM 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 PM CPU utilisation. This would require measuring the relationship between the number of vCPU 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 work, two different PMs (Host A and Host B) in a cloud testbed have been characterised with regression models, as shown in Figure 5 and 6. This experiment was carried out on a local Cloud Testbed by stressing the CPU to its full capacity using the *Stress-ng* tool [23]. More details on the experimental set up are found in Section 6. A linear regression model has

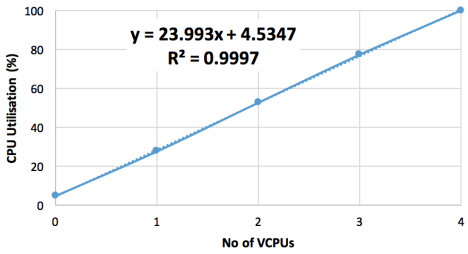

Figure 5: Number of vCPUs vs CPU Utilisation for Host A.

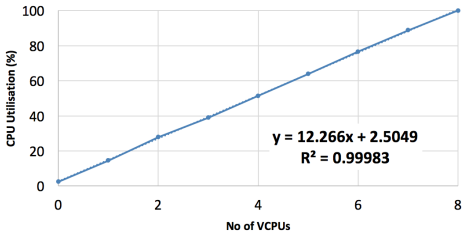

Figure 6: Number of vCPUs vs CPU Utilisation for Host B.

been applied to predict the PM CPU utilisation based on the used ratio of the requested number of vCPU for the VMs with consideration of its current

workload as the PM may be running other VMs already. The following Equation is used (4):

$$PMx_{PredUtil} = (\alpha \times (\sum_{y=1}^{VMCount} (VMy_{ReqvCPUs} \times \frac{VMy_{PredUtil}}{100})) + \beta) + (PMx_{CurrUtil} - PMx_{IdleUtil}) \quad (4)$$

$PMx_{PredUtil}$ is the predicted PM CPU utilisation; α is the slope and β is the intercept of the CPU utilisation. The $VMy_{ReqvCPUs}$ is the number of requested vCPU for each VM and $VMy_{PredUtil}$ is the predicted utilisation for each VMs. The $PMx_{CurrUtil}$ is the current PM utilisation and $PMx_{IdleUtil}$ is the idle PM utilisation.

4.3. PM Energy Consumption Prediction

After predicting the PMs workload, the third step is to predict the PMs power consumption based on the correlation of this predicted 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 7 and 8. Therefore, the PMs predicted

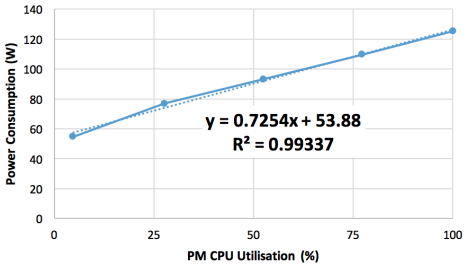

Figure 7: CPU Utilisation vs Power Consumption for Host A.

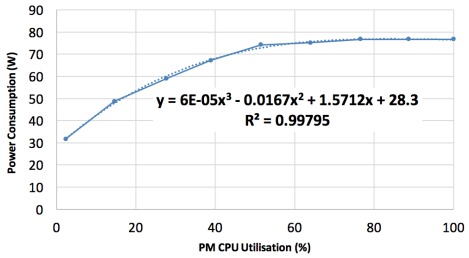

Figure 8: CPU Utilisation vs Power Consumption for Host B.

power consumption, $PMx_{PredPwr}$ measured by Watt, can be identified using a linear relation with the predicted PMs CPU utilisation, as shown in Figure 7 and in Equation (5). α and β are the slope and interceptor values obtained from the regression relation.

$$PMx_{PredPwr} = (\alpha \times (PMx_{PredUtil}) + \beta) \quad (5)$$

However, not all existing PMs necessarily follow a linear power model with their CPU utilisation, since the PMs are heterogeneous in nature, as shown

for example in Figure 8. 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 (6).

$$PMx_{PredPwr} = (\alpha(PMx_{PredUtil})^3 - \gamma(PMx_{PredUtil})^2 + \delta(PMx_{PredUtil}) + \beta) \quad (6)$$

Where α, γ and φ are all slopes, β is the intercept and $PMx_{PredUtil}$ is predicted PM CPU utilisation.

4.4. VM Energy Consumption Prediction

The fourth step of this framework is to attribute the predicted PMs energy consumption to the new requested VM and to the VMs already running on the physical host based on the energy-aware model introduced in Section 3. Hence, the predicted power consumption for the new VM, $VMx_{PredPwr}$, can be identified for the next interval time using Equation (7).

$$VMx_{PredPwr} = PMx_{IdlePwr} \times \left(\frac{VMx_{ReqvCPUs}}{\sum_{y=1}^{VMcount} VMy_{ReqvCPUs}}\right) + (PMx_{PredPwr} - PMx_{IdlePwr}) \times \left(\frac{VMx_{(PredUtil \times ReqvCPUs)}}{\sum_{y=1}^{VMcount} VMy_{(PredUtil \times ReqvCPUs)}}\right) \quad (7)$$

Where $VMx_{PredPwr}$ is the predicted power consumption for one VM measured by Watt. $VMx_{ReqvCPUs}$ is the requested number of vCPU and $VMx_{PredUtil}$ is the predicted VM CPU utilisation. $\sum_{y=1}^{VMcount} VMy_{PredUtil}$ is the total of vCPU for all VMs on the same PM. The $PMx_{IdlePwr}$ is idle power consumption and $PMx_{PredPwr}$ is the predicted power consumption for a single PM.

4.5. VM Total Cost Estimation

The final step in this framework is to estimate the total cost of the VM based on the predicted VM resource usage and the predicted VM power consumption. The energy providers usually charge electricity by the Kilowatt per hour (kWh). Therefore, converting power consumption to energy is required using the following Equation (8):

$$VMx_{PredEnergy} = \frac{VMx_{AvgPredPwr}}{1000} \times \frac{Time_s}{3600} \quad (8)$$

To estimate the total cost for the VM [14]. The following Equation is used (9):

$$\begin{aligned} VMx_{EstTotalCost} = ((VMx_{ReqvCPUs} \times \frac{VMx_{PredUtil}}{100}) \\ \times (CostpervCPU \times Time_s)) \\ + (VMx_{PredRAMUsage} \times (CostperGB \times Time_s)) \\ + (VMx_{PredDiskUsage} \times (CostperGB \times Time_s)) \\ + (VMx_{PredNetUsage} \times (CostperGB \times Time_s)) \\ + (VMx_{PredEnergy} \times CostperkWh) \end{aligned} \quad (9)$$

Where $VMx_{EstTotalCost}$ is the estimated total cost of the VM. The $VMx_{ReqvCPUs}$ is the number of requested vCPUs for each VM and $VMx_{PredUtil}$ is the predicted utilisation for each VM times the cost for the requested vCPUs for a period of time. $VMx_{PredRAMUsage}$ is the predicted resource usage of RAM times the cost for that resource for a period of time and so on for each resource such as CPU, disk and network. $VMx_{PredEnergy}$ is the predicted energy consumption of the VM times the electricity price as announced by the energy providers.

5. Energy-Aware Pricing Schemes

Cloud IaaS/PaaS providers mainly charge for their resources which come in the form of VMs with specific performance characteristics on the basis of fixed rates per unit of time. The rate levels depend on specific VM characteristics, such as CPU speed, network bandwidth, memory and storage space. At the same time, applications take decisions which can have an important impact on both energy consumption and performance. An example of such a decision is the level of parallelism in the event of multiple tasks scheduled on many different VMs. The application has the choice of the parallel execution of a number of tasks on many different VMs instead of using only a few, which may incur unnecessarily high energy costs by requiring a large number of physical servers to host the VMs. These increased energy costs are carried over to increased IaaS/PaaS prices and so lower profit levels for the providers.

One candidate solution could be the adoption of energy-aware pricing by the cloud providers in order to provide the necessary incentives to the customers for achieving a more efficient resource usage. Under such a scheme

the applications will be aware of the economic impact of their decision and so they will have the incentive to take energy costs into account, e.g., when they decide on the level of parallelism. Indeed, task scheduling at the application level may be more energy and performance effective than server consolidation by the IaaS/PaaS providers, since it is the applications which know what should be run in parallel and what should not.

However, additional information need to be provided by the existing infrastructure (e.g., energy consumption monitoring) to support such schemes. In response, the *Pricing Modeller* component is responsible for providing energy-aware price estimation and billing related to the operation of applications or VMs associated with them, see Figure 3.

The previous sections have focused on aspects related to the prediction of the energy consumption, as well as the resulted cost. As a next step, innovative energy-based pricing schemes are proposed, which were initially proposed in [24].

Static pricing: In this scheme, the price does not depend on energy consumption and depends only on VM characteristics, i.e.,

$$p = \frac{1}{T} \int_0^T p_{static}\left(VM, t\right) dt \tag{10}$$

Where VM is a parameter identifying the characteristics of the VM and $p_{static}\left(VM, t\right)$ is the static price of VM at time t. If the static price does not vary in time, i.e., $p\left(VM, t\right)$ is constant in the time parameter t, then no time averaging is necessary.

Two-part tariff: The actual form of IaaS price is comprised by two parts: a fixed one, depending only on static information of a VM, and a dynamic one, which depends on the average power usage. In a simple scheme, we consider a fixed part based on the static VM characteristics, plus the average power usage multiplied with the price per Watt-hour (Wh). Thus, the price p of a VM (starting at time 0 and up to time T) is computed by the formula

$$p = \frac{1}{T} \int_0^T p_{static}\left(VM, t\right) dt + \frac{1}{T} \int_0^T p_{energy}\left(t\right) W\left(t\right) dt \tag{11}$$

Where $p_{energy}\left(t\right)$ is the energy price at time t, and $W\left(t\right)$ is the power usage of the VM at time t.

Two-part tariff with energy-savings discounts: A disadvantage of the dynamic usage price is that the actual energy that an application may

use is not known by the developers at the time the SLA is established. A simple alternative is to pay a lump sum and then apply a discount based on the actual power consumption. Hence, the following two-part price can be used: α is a fixed price based on static info of a VM which also incorporates energy costs through the historical average power consumption or based on the prediction mechanisms presented in the previous sections, and b is a price discount depending on the level of power savings below the historical average or prediction. In this way it is not possible to pay more than the lump sum initial payment. More specifically, the price p is computed by the formula:

$$p = \frac{1}{T}\int_0^T p_{static}(VM,t)dt + \min\left(\frac{1}{T}\int_0^T p_{energy}(t)W(t)dt - \frac{1}{T}\int_0^T p_{energy}(t)W_{nominal}dt, 0\right) \quad (12)$$

where, $W_{nominal}$ is the nominal average power consumption, i.e., the power consumption already accounted for in the static price. Any average power consumption above $W_{nominal}$ does not increase price above the (time average) static price. Deviations below $W_{nominal}$ result into a proportional discount. The function $\min(x_1, x_2, \ldots)$ yields the numerically smallest of the x_i.

Linearly increasing pricing: The two-part tariff and energy-saving discounts pricing schemes assume that the price of energy could potentially vary in each epoch. However, such schemes do not consider any direct relation between the energy price and the total energy consumption. For example, an energy provider would reasonably like to avoid facing energy consumption bursts (e.g., during summer). Most of the energy providers usually provide a lower price per energy unit during the less burst periods (e.g., day / night). Motivated by this approach, the price per energy unit based on the total consumed energy in considered to be a linear increasing function. Other approaches (e.g., exponential function) may be also applied, in order to capture the notion of setting higher price per energy unit, as more energy is consumed during an epoch. The slope of the charging function will be set by the IaaS provider based on the factors consisting his own cost function (e.g., charging scheme or/and SLAs between IaaS and energy provider). For the linear assumption, p_{energy} can be written as $cW(t)$, assuming that c is a constant parameter set by the IaaS provider, showing how aggressively p_{energy} will increase with respect to the total energy consumption. In order to prevent

IaaS provider to charge arbitrarily high prices, an upper bound is set, such that $cW(t) \leq p_{energy.upper}$. Thus, the price p is computed by the formula:

$$p = \frac{1}{T}\int_0^T p_{static}(VM,t)dt + \min\left(\frac{1}{T}\int_0^T cW^2(t)dt, \frac{1}{T}\int_0^T p_{energy_upper}(t)W(t)dt\right) \quad (13)$$

95th percentile rule-based pricing: The 95th percentile rule is a widely used pricing scheme in telecommunications for charging the transit traffic sent by lower-tier ISPs. By employing this scheme, transit ISPs intend to penalise lower-tier ISPs in case of traffic bursts. A similar pricing scheme could be employed by IaaS providers for penalising bursts of the consumed energy. To implement this scheme, it is assumed that the energy consumption within the infrastructure of an IaaS provider is measured or sampled and recorded (e.g., log file). At the end of each billing cycle (e.g., every month), the energy consumption samples are sorted from highest to lowest, and the top 5% of data is thrown away. The next highest measurement is the 95th%, and the customer will be billed based on that energy consumption. Let $l^*(t)$ denote the 95th% measurement of the energy consumed by the customer at time t. l^* is then defined as $max\{l \mid P(W > l) \geq 0.05\}$. Thus, the price will be:

$$p = \frac{1}{T}\int_0^T p_{static}(VM,t)\,dt + \frac{1}{T}\int_0^T p_{energy}(t)\,l^*(t)\,dt \quad (14)$$

6. Experimental Set Up and Design

This section describes the environment and the details of the experiments conducted in order to evaluate the work presented in this paper. In terms of the environment, the experiments have been conducted on the Leeds Cloud testbed. The details of this testbed and the experiments will be discussed next.

6.1. Cloud Testbed

The cloud testbed consists of a cluster of commodity Dell servers, and each one of these servers has Centos version 6.6 installed as its operating system (OS). Two of these servers, one with a four core X3430 Intel Xeon

CPU (Host A) and the other with an eight core E3-1230 V2 Intel Xeon CPU (Host B), have been used for the experiments presented in this paper. Also, each server has a total of 16GB of RAM and 250GB of SATA HDD. Additionally, the testbed has a Network File System (NFS) share running on the head node of the cluster and providing a 2TB total storage for VM images.

The architecture of this testbed is shown in Figure 9. The testbed utilises OpenNebula [25] version 4.10.2 as the Virtual Infrastructure Manager (VIM). For the Virtual Machine Monitor or Manager (VMM), the KVM hypervisor is used [26] hypervisor.

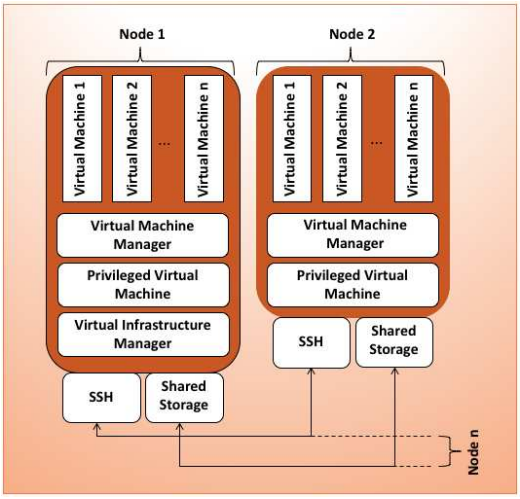

Figure 9: Cloud Testbed Architecture.

6.2. Monitoring Infrastructure

The resources usage and energy monitoring on the Cloud testbed is shown on Figure 10. At the physical host level, each PM has a WattsUp [27] meter attached to directly measure power consumption at per second basis for each PM. The measured power consumption are then pushed to Zabbix [28], which is used for resources usage monitoring purposes.. Additionally, Zabbix also monitors the resources usage, like CPU, memory, network and disk, for each of the running PMs and VMs.

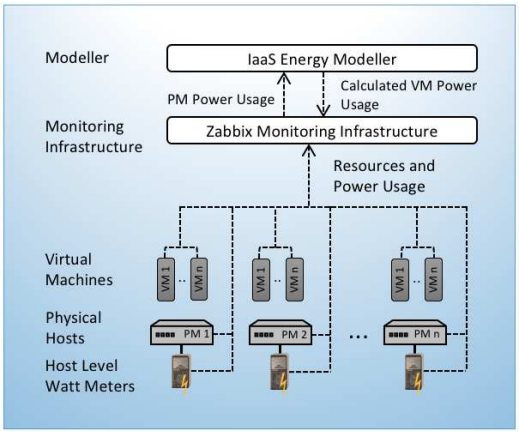

Figure 10: Monitoring Infrastructure.

6.3. Design of Experiments

The aim of the experiments is to demonstrate that: 1) the energy-aware model is capable of fairly attributing the PMs energy consumption to homogeneous and heterogeneous VMs, and 2) the energy-aware cost prediction framework is capable of predicting the workload and power consumption as well as estimating the total cost of the VMs at service operation based on historical static and periodic workload patterns.

The size of the VM is identified by its capacity in terms of the number of vCPUs and memory size. For example, if two VMs have the same number of vCPUs on each, then they are considered homogeneous VMs. If one has one vCPU and the other has two or more vCPUs, then they are considered heterogeneous VMs. Rackspace [29] is used as a reference for the VMs configurations. The experiments consider three sizes of VMs, VM_A(small), VM_B(medium) and VM_C(large) are provided with different capacities. The VMs are allocated with 1, 2 and 3 vCPUs, 1, 2 and 3 GB RAM, 10 GB disk and 1 GB network, respectively. The cost of the virtual resources are set according to ElasticHosts [30] and VMware [31] prices are followed: where 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 [32].

In terms of evaluating the energy-aware model, the first experiment is designed to run VM_A(small), VM_B(medium) and VM_C(large) on a PM (Host A), and to run the same types of these three VMs on a different PM

(Host B). The aim of this experiment is also to explore how the energy consumption is attributed to the same types of VMs when being run on different PMs. The software tool *Stress-ng* is used along with *cpulimit* to generate synthetic workload on the VMs at any level of CPU utilisation. All the VMs used in this experiment are designed to be idle for 15 minutes at the first stage, and then actively run at 80% of CPU utilisation for another 15 minutes at the second stage. This way can help to explore how the idle and active power consumption of the PM are attributed to the VMs over time. All the experiments are repeated five times and the statistical analysis is performed to consider the mean values of the results and eliminate any anomalies.

To evaluate the energy-aware cost prediction framework, a number of experiments have been conducted on the testbed to synthetically generate historical workload data. The historical data has been generated to represent real workload patterns of cloud applications, including static and periodic, by stressing all the resources (CPU, memory, disk and network) on different types of VMs with the *Stress-ng* tool. The generated workload of each VM type has four-time intervals of 30 minutes each. The first three intervals will be used as the historical data set for prediction, and the last interval will be used as the testing data set to evaluate the predicted results. The prediction process works offline by firstly predicting the VM workload using the ***auto.arima*** function in R package [22] to automatically select the best fit model of ARIMA based on AIC or BIC value. Once the VM workload is predicted, the process is then completed by going through the steps of the introduced framework to consider the correlation between the physical and virtual resources and consequently predict the power consumption and then estimate the total cost of the VMs running on different PMs

7. Evaluation and Discussion

This section presents the evaluation of the energy-aware model and the energy-aware cost prediction framework. The figures below show the predicted results for three types of VMs, VM_A(small), VM_B(medium) and VM_C(large), each instance running on two different PMs based on a historical periodic workload pattern. Because of space limitation, only VM_A(small) and VM_C(large) results are shown.

7.1. Energy-Aware Virtual Machine Model

The conducted experiment shows the results of energy consumption attribution to heterogeneous VMs running on a PM (Host A). Additionally, this experiment also presents the results of attributing the same types of VMs on another PM (Host B).

7.1.1. Host A

The mean power consumption and CPU utilisation for VM_A(small) and VM_C(large) running on Host A are shown in Figures 11 and 12, 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.

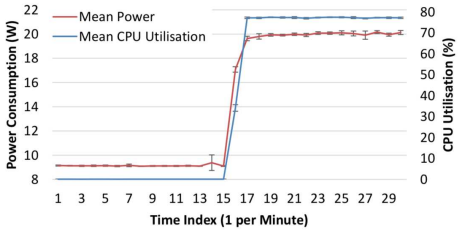

Figure 11: Mean Power Consumption and CPU Utilisation for VM_A(small).

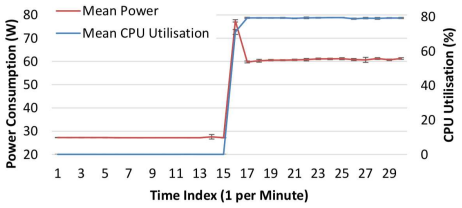

Figure 12: Mean Power Consumption and CPU Utilisation for VM_C(large).

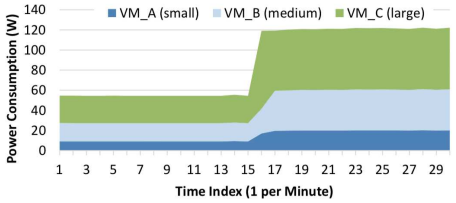

Figure 13: PM Mean Power Consumption Attributed to each VM.

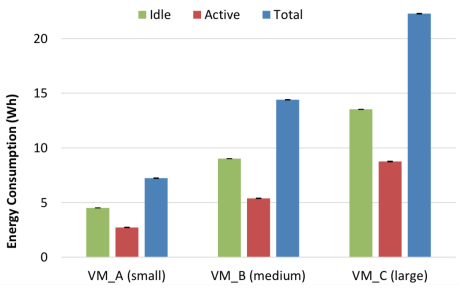

Figure 14: Mean Energy Consumption per VM (for 30 minutes).

Figure 13 shows the distribution of the PMs mean power consumption to all these three VMs over time, and Figure 14 shows the mean energy consumption per VM in terms of their idle, active and total energy. As the VMs are heterogeneous, therefore have different attribution of the idle and active energy consumption, which fairly corresponds to their size. The energy consumption of VM_A(small) is about two times smaller than VM_B(medium)

and three times smaller than VM_C(large), which is fairly based on their CPU utilisation and sizes defined by the number of vCPUs each VM has.

7.1.2. Host B

The mean power consumption and CPU utilisation for VM_A(small), and VM_C(large) running on Host B are shown in Figures 15 and 16, respectively. Recall, all of the VMs are idle in the first 15 minutes and actively running with 80% of CPU utilisation for the remaining 15 minutes. Figure 17 shows the distribution of the PMs mean power consumption to all three VMs, and Figure 18 shows the mean energy consumption per VM in terms of their idle, active and total energy. As the VMs are heterogeneous in terms of the size, they consequently have different attribution of the idle and active energy consumption. The energy consumption of VM_A(small) is about two times smaller than VM_B(medium) and three times smaller than VM_C(large), which is fairly based on their CPU utilisation and sizes defined by the number of vCPUs each VM has.

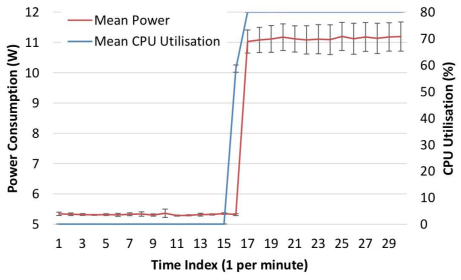

Figure 15: Mean Power Consumption and CPU Utilisation for VM_A(small).

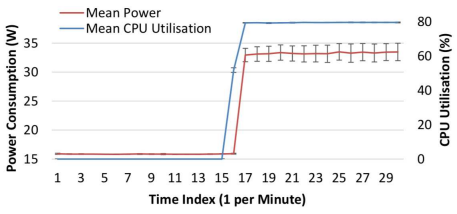

Figure 16: Mean Power Consumption and CPU Utilisation for VM_C(large).

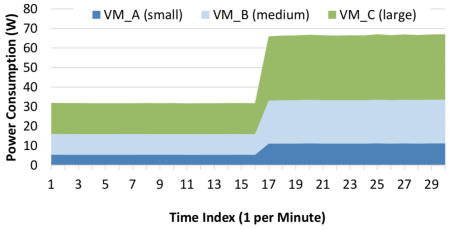

Figure 17: PM Mean Power Consumption Attributed to each VM.

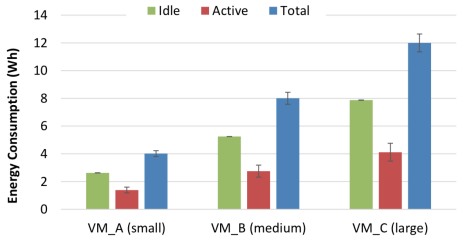

Figure 18: Mean Energy Consumption per VM (for 30 minutes).

The conducted experiment has 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 14 and 18. Hence, enabling energy-awareness at the VM level can help the cloud service providers to monitor the energy consumption of the VMs and, if necessary, migrate the VMs to another host to maintain their energy goals.

Further, the conducted experiment has 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, as will be discussed next.

7.2. Energy-Aware Cost Prediction Framework

The conducted experiment shows the prediction results for three types of VMs, VM_A(small), VM_B(medium) and VM_C(large), based on static and periodic workload patterns 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 mix of VMs with a mix of workload patterns when being run on different PMs.

In terms of the historical and testing data sets, Figures 19 and 21 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 static and periodic workload and give a very accurate prediction accordingly. The predicted VMs disk and network workload are also matching 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 pattern for the RAM and CPU usage are closely matched in each interval. Beside the

predicted mean values, the figures also show the high and low 95% and 80% confidence intervals.

Table 1: Prediction Accuracy for VM_A(small).

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
Power Consumption Host A	0.010496	0.10504	0.045515	0.029576	0.11785
Power Consumption Host B	0.010079	0.11109	0.049255	0.017091	0.07599

Based on the predicted workload for each VM, their power consumption is predicted via the remaining steps within the framework. Figures 20 and 22 show the predicted versus the actual results of the power consumption for VM_A(small), VM_B(medium), and VM_C(large) when being run on Host A and Host B, noting that the Host B is more energy efficient 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 the VMs, hence the predicted power consumption of all the VMs is closely matched the pattern of the predicted VMs CPU utilisation, as shown in Figures 19 and 21.

In terms of prediction accuracy, a number of metrics have been used to evaluate the predicted workload (CPU, RAM, disk, network) and power consumption for the VM_A(small), and VM_C(large) based static and periodic workload pattern as presented on Tables 1 and 2 respectively. These metrics include, *Absolute Percentage Error (APE)* which measures the absolute value of the ratio of the error to the actual observed value; *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*

Table 2: Prediction Accuracy for VM_C(large).

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
Power Consumption Host A	0.026211	0.20869	0.095949	0.010313	0.11750
Power Consumption Host B	0.000131	0.16633	0.062928	-0.03101	0.13774

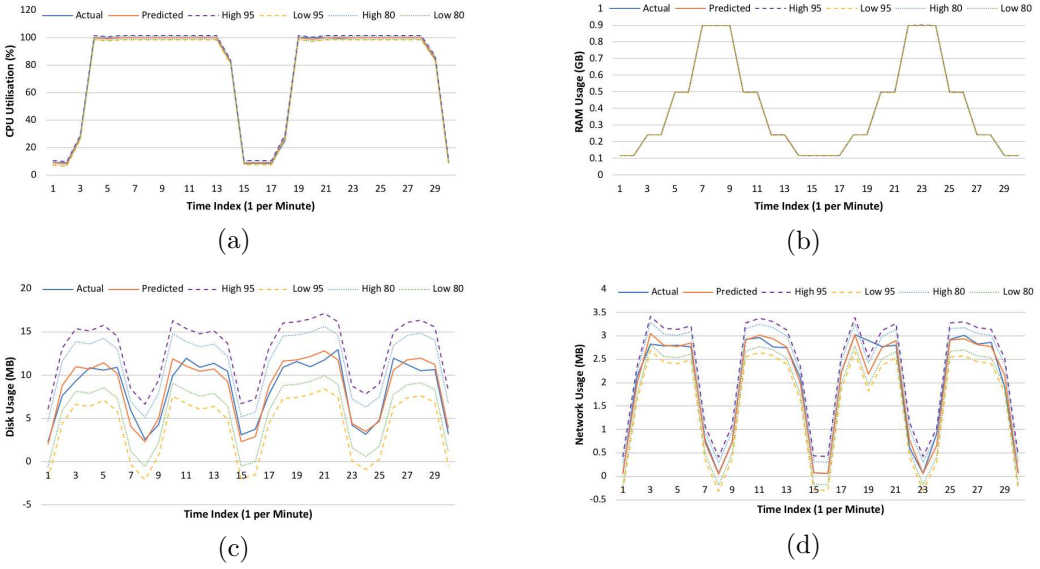

Figure 19: The prediction Results for VM_A(small) (for 30 minutes).

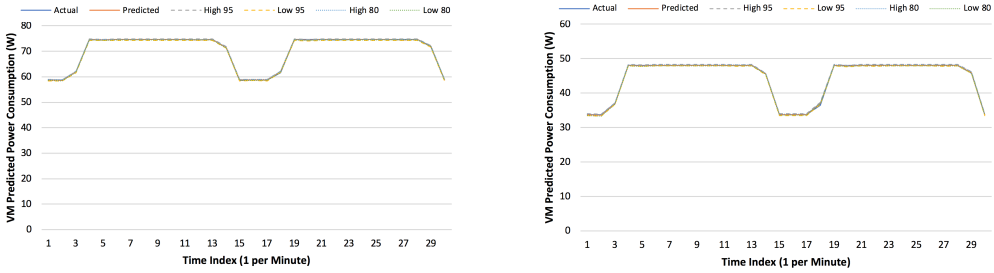

Figure 20: The prediction Power Consumptionfor VM_A(small) (for 30 minutes).

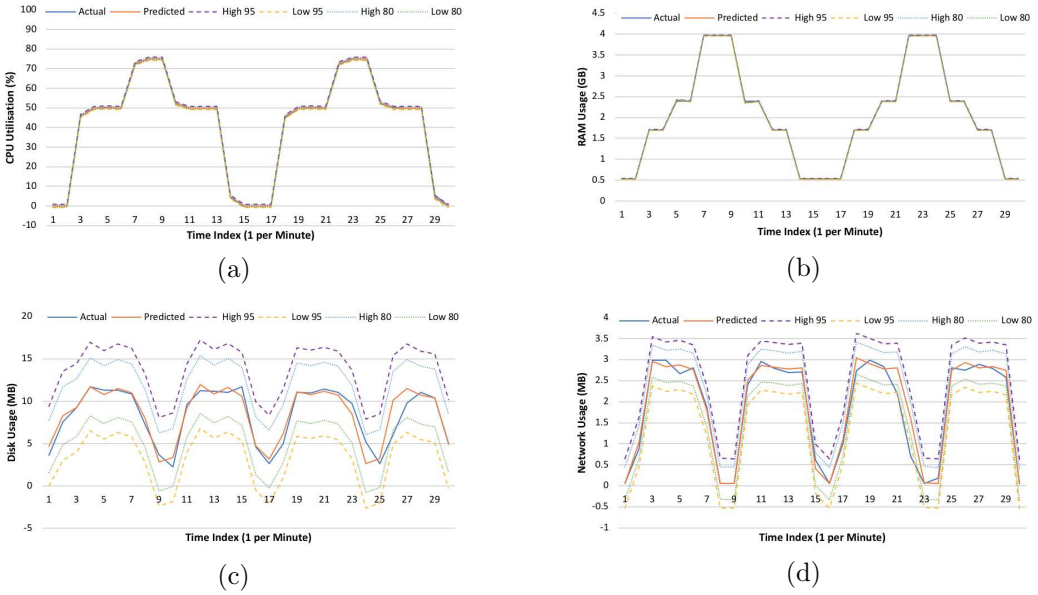

Figure 21: The prediction Results for VM_C(large) (for 30 minutes).

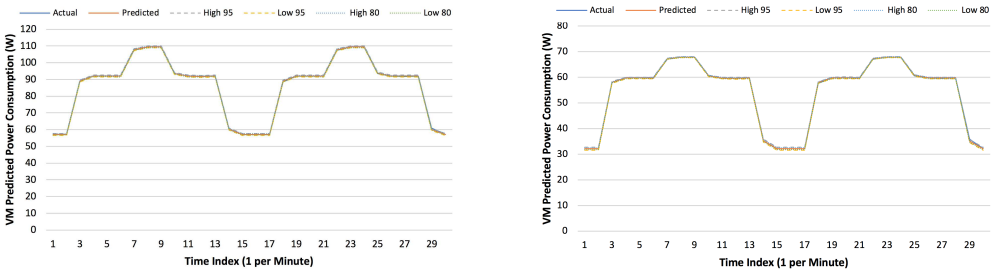

Figure 22: The prediction Power Consumptionfor VM_C(large) (for 30 minutes).

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 [33].

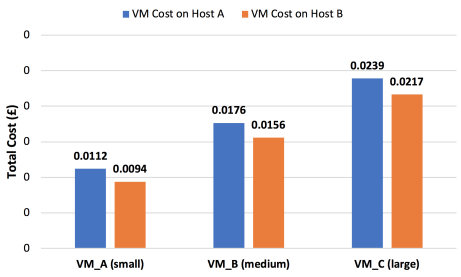

Figure 23: The predicted VMs total cost on Host A and Host B.

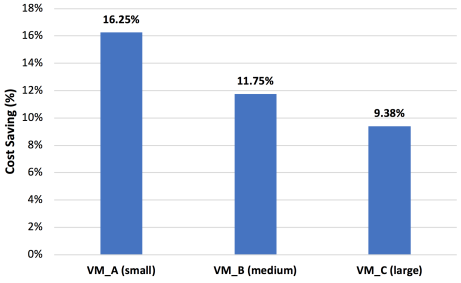

Figure 24: The VMs Cost Saving on Host B.

This framework is also capable of estimating the total cost for three types of VMs hosted/running on two different PMs as shown in Figure 23, which presents the estimated total cost of the VM_A(small), VM_B(medium), and VM_C(large) running on different PMs (Host A and Host B). As the VMs are heterogeneous, therefore the costs of VMs are different. The cost of VM_A(small) is about two times smaller than VM_B(medium) and three times smaller than VM_C(large) when there are running on Host B, which is fairly based on their 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 comparing to Host A as shown in Figure 24.

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

7.3. Pricing Schemes Evaluation

The goal of this analysis is to compare the economic implications of the choice of pricing scheme by a service provider. In particular, the static and energy-based pricing schemes presented in the previous section are compared. To do this, models of cloud service providers sharing the same capabilities and the same cost structure are considered, their only difference being the

pricing scheme adopted by each. The basic model assumptions are briefly presented. The detailed assumptions of our model are presented in [24].

IaaS/PaaS provider: it is assumed that the PaaS layer is offered by the same economic entity, which offers the IaaS. Thus, whenever the reference to IaaS means the combination of IaaS/PaaS. Each IaaS/PaaS provider has an infinite number of physical servers at his disposal. Each server is populated by VMs belonging to possibly different applications and the CPU speed is split equally among the VMs. The provider is able to freely scale, i.e., the server consolidation policy is such that the number of active physical servers scales in proportion to the number of VMs in the infrastructure. A two-part tariff adopted by the provider if further considered.

User demand for application requests: Each application has a different throughput demand (rate of instructions or requests to be executed at the VMs of this application), which decreases to zero if the average processing delay of each instruction/request becomes too high. It is assumed that the benefit decreases as response delay increases. If the delay becomes too high, the benefit will become negative and requests will start balking at this point.

Applications: It is assumed that each application is employing a number of VMs. The cost for the SaaS provider of this application depends on the parameters of the two-part tariff employed by the IaaS/PaaS provider, while its revenue is based on the number of the completed requests (e.g., euros/request). The application decides how many VMs to buy from a particular IaaS provider such that its profit is maximised.

Hence, the economic quantities considered are:

- The level of profits for each type of provider.
- The level of payments made by the customers of each provider type.
- The level of overall satisfaction of the customers of each provider type. Since the comparison depends on the market structure; the actions of service providers under two extreme cases are considered, which are i) *monopoly*, and ii) *perfect competition*.

The main outcomes of the analysis are summarised below:

Incentive to adopt energy-aware IaaS/PaaS layers under monopoly: The profit of IaaS providers in a monopoly increases if a two-part tariff incorporating energy costs is used, compared to a static pricing scheme. Thus,

IaaS/PaaS providers have the incentive of adopting energy-aware IaaS layer regardless if the upper layers exist or not.

Incentive to adopt energy-aware IaaS/PaaS layers under competition: The two-part pricing scheme incorporating energy costs is a viable strategy under competition: the IaaS/PaaS providers obtain a share in a competitive market. This is not true under the static pricing scheme, where an IaaS/PaaS provider cannot have a non-zero market share and cover his costs at the same time. This result implies that SaaS providers are more profitable using IaaS/PaaS providers which charge according to energy consumption. Thus, SaaS providers will be attracted to energy-aware IaaS/PaaS providers even though the former are not aware of the energy used by their components; their sole criterion being the resulting price.

Incentive to adopt energy-aware SaaS layer: SaaS providers obtain greater profits when they become energy-aware in a market of competitive energy-aware IaaS/PaaS. This is done through application-level scheduling of more energy consuming requests on VMs residing in more power efficient hosts. As a result, IaaS providers continue to be better off using the two-part tariff even after SaaS providers start being energy-aware.

The first part of our analysis considers whether ***energy-awareness of IaaS/PaaS providers*** is profitable for IaaS/PaaS and ***non-energy-aware SaaS providers***. A non-energy-aware SaaS provider means that decisions are not taken on the basis of energy consumption. For example, a SaaS provider is not able to decide which tasks to schedule on which VMs. However, it decides which IaaS/PaaS provider to use on the basis of total price charged.

In our first scenario, it is assumed that the IaaS/PaaS provider operates in a **monopolistic market**, where two different applications (in terms of quality of service characteristics) request for IaaS/PaaS services. The profits of a monopolistic IaaS/PaaS provider employing: i) *a two-part tariff*, which incorporates energy consumption, and ii) *a static price* are numerically evaluated. Figure 25 depicts the profits as a function of the maximum average request response delay tolerated by the users of application 1 (normalised by the maximum tolerated delay for application 2). The profits brought by the two-part tariff are always greater than those brought by the static pricing scheme. They coincide only if the QoS characteristics of the two applications are the same. The greater the diversity between the applications, the greater the difference in profits.

The second scenario considers the case of **perfect competition** among

two IaaS/PaaS providers. Under perfect competition without entry costs, no IaaS/PaaS provider is able to make strictly positive profits because in that case no demand will be available. This is because the demand is attracted by other providers, which choose to operate at a smaller albeit non-zero profit margin by slightly reducing their prices. Thus at market equilibrium, competitive IaaS/PaaS providers obtain zero profits and barely cover their costs. Since the interest is the comparison of the effect of the pricing scheme on competition, IaaS/PaaS providers are compared under the same characterising parameters (including maintenance and energy costs) except those concerning their pricing scheme.

As an exposition of the competition between IaaS/PaaS providers and the effect of the pricing scheme, consider an example which examines the profits of two applications as a function of their diversity. The users of the applications are assumed not to tolerate average request response delays above some value, which is specific to each application. Figure 26 depicts the payments per time unit incurred by each application under two different pricing schemes: i) *a two-part tariff*, which incorporates energy consumption, and ii) *a static price*. The horizontal axis represents the maximum tolerable delay by users of application 1 (normalised to that of application 2).

For stringent delay requirements (max tolerable delay is less than 0.3), application 1 does not at all use the IaaS/PaaS provider with static pricing since the high costs outweigh benefits. The latter hosts application 2 only, at a competitive price. When the delay requirements of application 1 are not so stringent, the demand rises and application 1 starts using the static IaaS/PaaS provider, but at a cost which is not competitive: application 1 payments exceed the ones offered by the IaaS/PaaS provider employing a two-part tariff. For values of the max tolerable delay above 1, the less tolerable users belong to application 2 now, and they bare most of the costs in both IaaS/PaaS providers. Nevertheless, the static IaaS/PaaS provider continues not to be competitive as the payments resulting for application 2 exceed those by the IaaS/PaaS provider employing the two-part tariff.

The second part of the analysis considers whether energy-awareness of SaaS providers is economically sensible. In order to make the effects of energy-awareness clearly visible, the model is refined to allow for i) *physical hosts with different power efficiency*, ii) *requests with different energy consumption*.

Two types of hosts are considered. Both host types consume the same power while their CPU idles. While active, *type 1 host* is more power efficient.

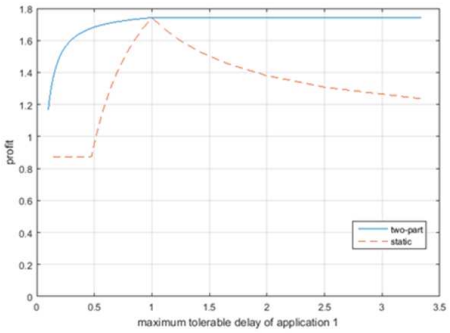

Figure 25: IaaS/PaaS provider profits in a monopoly using a two-part tariff incorporating energy charges (solid curve) and a static price (dashed).

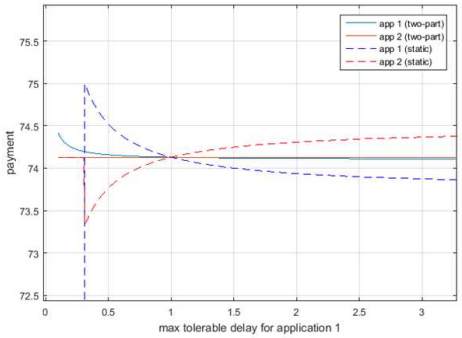

Figure 26: Comparison of payments by two applications to IaaS/PaaS providers as a function of application QoS diversity.

The fact that *type 1 hosts* are more power efficient has an implication for the VM scheduling policy of the IaaS/PaaS provider. Since the latter strives to have minimal energy costs, more power efficient hosts are preferred to less efficient ones. Thus, the VM scheduling will try to allocate *type 1 hosts* first to meet demand; *type 2 hosts* will be used only if it is not possible to meet demand only by utilising *type 1 hosts*. This is under the assumption that the VM scheduling algorithm is allowed to freely reallocate all VMs on the available hosts. It is further assumed a unit rate of *type 1 requests* consumes $w_1 > 1$ times the one of *type 2*. The precise power consumption depends on the host type the request is executed.

As a next step, the implications in power consumption due to the application being energy-aware or not is considered. First the "legacy" case is looked at, where an application has no information about the power consumption of its components. In this case, the application cannot differentiate between the more and less energy consuming request types. Moreover, it cannot have information about the energy efficiency of its VMs. Thus the requests are scheduled on VMs independently of their type. Let us now consider how an energy-aware application allocates requests on its VMs. Since *type 1 hosts* are more power efficient and *type 1 requests* are more energy consuming (as $w_1 > w_2$), an energy-minimising scheduling policy ought to place *type 1 requests* on *type 1 hosts* and use *type 2 hosts* only if necessary or for serving (the less consuming) *type 2 requests*. In the ***monopoly*** scenario, the SaaS provider intends to optimise the number of VMs requested by the applica-

tion, while the IaaS/PaaS provider chooses the optimal price to maximise profits. In Figure 27, the above problem is numerically solved and the maximum profits for the monopolist as a function of the number of power efficient (namely type 1) hosts H_1 is depicted. π_e is the energy price of the energy provider. As the number of *type 1 hosts* increases, the energy-saving effect of the scheduling of requests performed by the application becomes more significant. The upward slopes for the energy-aware SaaS providers (the two solid curves) decrease around $H_1 = 26$. This is the point where *type 1 hosts* serve exclusively *type 1 requests*. For greater values of H_1, *type 2 requests* are served by *type 1 hosts* and hence the savings effect is less pronounced. Beyond $H_1 = 55$ there is no profit difference as all requests are served by *type 1 hosts* and request scheduling does not have any effect, since VM scheduling makes sure only the power efficient hosts are utilised. We consider different values of π_e (0.05 and 0.01 correspondingly) in order to investigate how the profits of the cloud providers are affected by the energy price of the energy provider.

In the ***perfect competition*** scenario, the IaaS/PaaS providers have zero profit margin. Applications however have strictly positive profits and it is observed that their profits increase by being energy-aware. Again, the SaaS provider intends to maximise profits. One can move from the legacy allocation of *type 1 requests*, where these are distributed equally among all VMs (irrespective of the host they are running on), to the allocation produced by energy-awareness, by shifting small loads of *type 1 requests* that reside on any VMs on *type 2 hosts* to VMs on *type 1 hosts*. Hence, application level energy-awareness increases applications' profits. Figure 28 presents the profits of energy-aware (solid curve) and "legacy" applications (dashed) in competitive markets for IaaS/PaaS, as functions of the proportion of high energy requests.

Based on the aforementioned analysis, it is concluded that applications themselves would want to adopt energy-based technologies because they become more profitable if IaaS/PaaS charge according to energy consumption.

8. Related Work

This section reviews existing work and categorises it into three lines of research: VM energy modelling, prediction modelling and pricing modelling in cloud computing.

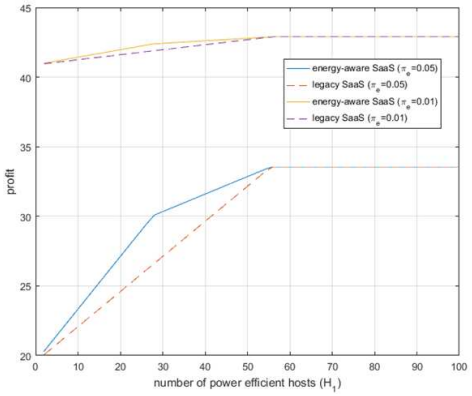

Figure 27: IaaS/PaaS provider profits in the case of monopoly as a function of the number of power efficient hosts.

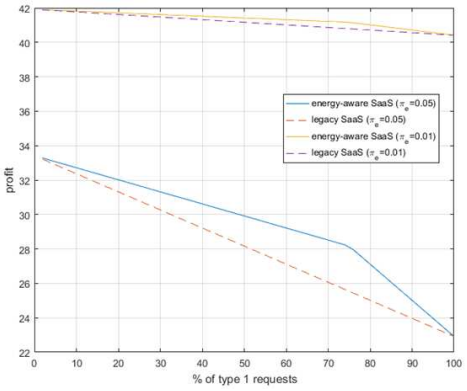

Figure 28: Profits of energy-aware and "legacy" applications as functions of the proportion of high energy requests.

8.1. VM Energy Models

Unlike PMs, VMs energy consumption cannot be measured directly as they do not have direct hardware interfaces to plug in any of the wall Watts meters. Therefore, their energy information can be indirectly identified via software tools that model the energy consumed by the PMs in which they are hosted [34] with the use of different approaches, like resource usage-based [35], [13], [12, 9] lookup table-based [36], and performance counters-based [8].

In terms of the PMs idle power consumption, most of the related work does not consider it or attributes it evenly to the VMs, which would not be fair when heterogeneous VMs are running alongside on the same PM. The only exception is the model presented in [35] which considers attributing the PMs idle power consumption to homogeneous and heterogeneous VMs; yet when part of the PMs CPU and memory resources are assigned to the VMs, it only attributes part of the PMs idle power to VMs, which is considered unfair as that given PM is switched on to run and maintain the status of the VMs; otherwise, that given PM could be switched off to save its idle power consumption. In terms of the PMs active power consumption, some of the related work models [36, 35, 12] attribute it to homogeneous VMs only. The other models [8], [9] consider attributing the PMs active power to homogeneous and heterogeneous VMs, but using different approaches. The model introduced in [9] is the only model that has a similar approach to the one introduced in this research when attributing the PMs active power consumption to the VMs; however, their model still lacks fair attribution of

the PMs idle power consumption to heterogeneous VMs.

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

8.2. Energy Prediction Models

As stated in [37], predicting the energy consumption of cloud applications and VMs about to be deployed and run would require understanding the characteristics of the underlying physical resources, like idle power consumption and variable power under different utilisation of workload, and the projected virtual resources usage. Most of the existing work [20], [38, 39] introduced different approaches to predict the workload in order to meet the demand and efficiently provision the resources in cloud environments, yet not considering the energy consumption and energy efficiency of the resources. However, only the work presented in [40] considers predicting the workload and translating it into energy consumption in a cloud environment. The work presented in [40] is the only work that has a similar approach to the one introduced in this research in terms of predicting the workload and then translating it into energy consumption. Nonetheless, their approach is only focused at the PM level, whereas the prediction approach introduced in this paper focuses at both the VM and PM levels.

In terms of prediction based on historical data, predicting the resources usage, energy consumption and estimating total cost of the VMs, some of the related work [20] predict the workload only without consider the estimation of costs or energy consumption of the VMs. The other methods presented in [41, 42] consider total cost of the VMs including the cost of energy consumption based on (e.g. number of VMs and data size). Nonetheless, their objectives do not consider estimating the total cost or energy consumption.

The approach of the framework presented in this paper first predicts the workload of the VMs and then correlates the predicted VM workload with the PM to estimate the PMs workload and power consumption, from which

the power consumption for the VMs is predicted, then, estimated the VMs total cost accordingly.

8.3. Pricing Models

In the *pay-as-you-go* scheme the customer pays for the resources made use of. With this scheme, the customer can choose the amount of a variety of characteristics that will compose the VMs. The basic characteristics of the VMs are the capacity of the CPU, memory, storage, data transfer and operating system. One other popular scheme used is the *periodic payment* (e.g., monthly, semester, yearly subscriptions, etc.) or pre-payment. The customers pay or pre-pay the use of specific resources, having a discount on the hourly charges. Usually under these schemes, if the needs of the customer change, the resources reserved for him cannot be returned and the amount is not refunded. Another innovating scheme is *on-demand / reserved instances*, where the customers pay for compute capacity by the hour with no long-term commitments. The notion behind this scheme is the reservation of the resources before their use for a specific amount of time. A similar scheme is *spot instances*, where the customer buys the unused capacity and runs it until the price of the instances bought becomes higher than the actual bid. The spot price changes periodically based on supply and demand, and customers whose bids meet or exceed it, gain access to the available spot instances [43].

Pricing in cloud computing has been studied extensively in the past [44, 45, 46] and most approaches consist of a combination of a fixed or variable price per VM instance and an additional usage charge based on the actual use of computing resources such as CPU cycles, network bandwidth, memory and storage space. Our work in [45] does not focus on the economic implications of the proposed pricing scheme, while the work in [46] proposes a demand-response mechanism which the cloud employs to cope with the variability in electricity prices. In our recent work [24], a novel pricing scheme based on energy consumption of cloud resources is proposed. In [47], the economic implications of the choice of pricing schemes by an IaaS/PaaS provider are compared, as well as the incentives of SaaS providers to adopt an energy-aware framework.

9. Conclusion and Future Work

This paper has introduced a cloud system architecture and evaluated an energy-aware model that enables a fair attribution of a PMs energy consumption to homogeneous and heterogeneous VMs based on their utilisation and size, which reflect the physical resource usage by each VM. Also, it has proposed an energy-aware cost prediction framework that can predict the resource usage, power consumption and estimate the total cost for the VMs during the operation of cloud services. A number of direct experiments were conducted on a local Cloud Testbed to evaluate the capability of the prediction models. Overall, the results show that the proposed approach can fair attribution of a PMs energy consumption to the VMs and predict the resource usage, power consumption and estimate the total cost for the VMs with a good prediction accuracy based on Cloud workload patterns. Unlike other existing works, this approach considers the heterogeneity of VMs with respect to predicting the resource usage, power consumption and estimating the total cost.

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

Additionally, a set of novel energy-aware pricing schemes is proposed to enhance IaaS/PaaS providers choosing their optimal pricing strategy, reflecting also our target for incentivising the customers to be energy-efficient. The proposed pricing schemes differ in terms of aggressiveness with respect to the charging of energy consumption bursts. To this extent, a mathematical model of applications and IaaS/PaaS providers and show that applications which adapt to energy-based information and the proposed energy-based pricing schemes by appropriately scheduling requests to VMs, extract higher profits compared to being non-adaptive. Although the model is a gross simplification of reality, it is valuable in that it clearly shows the potential economic benefits for applications to respond to appropriate pricing signals. Thus, it is not only that applications become more power efficient once they utilise an energy-aware framework but they have an economic incentive to utilise it. The IaaS/PaaS providers are the likely first adopters of energy-aware layers as it increases their profits even when the application providers are

not energy-aware. Even if the aforementioned analysis shows that if SaaS providers adopt the energy-aware SaaS layer they will also see their profits increase, this does not mean that they will adopt an energy-aware framework as they have no means of evaluating the benefit of doing so.

Future work includes the extension of our approach and integrate it with performance prediction models to determine the costs of different scenarios. Besides, further investigation will focus on VM performance prediction models, dynamic placement of VMs, and demonstration of the trade-off between cost, power consumption and performance. Also, the scalability aspects with different prediction algorithms will be considered to further show the capability of the proposed work. Finally, 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 framework to predict the workload, power consumption and estimate total cost of the VMs based on different types of workload patterns.

Acknowledgment

The authors would like to thank the European Commission for supporting this work under FP7 contract 610874 (ASCETiC project) and Horizon 2020 contract 687584 (TANGO project).

References

[1] J. Conejero, O. Rana, P. Burnap, J. Morgan, B. Caminero, C. Carrión, Analyzing Hadoop power consumption and impact on application QoS, Future Generation Computer Systems 55 (2016) 213–223.

[2] A. Narayan, S. Member, S. Rao, S. Member, Power-Aware Cloud Metering, IEEE Transactions on Services Computing 7 (3) (2014) 440–451.

[3] A. Beloglazov, R. Buyya, Y. C. Lee, A. Zomaya, A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems Table of Contents, in: Advances in computers, Elsevier, 2011, pp. 47–111.

[4] K. Djemame, R. Bosch, R. Kavanagh, P. Alvarez, J. Ejarque, J. Guitart, L. Blasi, PaaS-IaaS Inter-Layer Adaptation in an Energy-Aware Cloud Environment, IEEE Transactions on Sustainable Computing 2 (2) (2017) 127–139.

[5] M. Bagein, J. Barbosa, V. Blanco, I. Brandic, S. Cremer, H. D. Karatza, L. Lefevre, T. Mastelic, A. Oleksiak, 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. 2 (2) (2015) 105–131.

[6] K. Djemame, D. Armstrong, R. Kavanagh, A. J. Ferrer, D. G. Perez, D. Antona, J.-c. Deprez, C. Ponsard, D. Ortiz, M. Macias, J. Guitart, F. Lordan, J. Ejarque, R. Sirvent, R. Badia, M. Kammer, O. Kao, E. Agiatzidou, A. Dimakis, C. Courcoubetis, L. Blasi, 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.

[7] F. Lordan, E. Tejedor, J. Ejarque, R. Rafanell, J. Alvarez, F. Marozzo, D. Lezzi, R. Sirvent, D. Talia, R. Badia, Servicess: An interoperable programming framework for the cloud, Journal of Grid Computing (2013) 1–25.

[8] H. Yang, Q. Zhao, Z. Luan, D. Qian, iMeter : An integrated VM power model based on performance profiling, Future Generation Computer Systems 36 (2014) 267–286.

[9] M. Zakarya, L. Gillam, An Energy Aware Cost Recovery Approach for Virtual Machine Migration, in: 13th International Conference on Economics of Grids, Clouds, Systems, and Services, 2016, pp. 175–190.

[10] P. Garraghan, I. S. Moreno, P. Townend, J. I. E. Xu, An Analysis of Failure-Related Energy Waste in a Large-Scale Cloud Environment, IEEE Transactions on Emerging Topics in Computing 2 (2) (2014) 166 – 180.

[11] W. Dargie, A stochastic model for estimating the power consumption of a processor, IEEE Transactions on Computers 64 (5) (2015) 1311–1322.

[12] R. Kavanagh, D. Armstrong, K. Djemame, Towards an Energy-Aware Cloud Architecture for Smart Grids, in: 12th International Conference on Economics of Grids, Clouds, Systems and Services, 2015, pp. 1287–1294.

[13] I. Alzamil, K. Djemame, D. Armstrong, R. Kavanagh, Energy-Aware Profiling for Cloud Computing Environments, Electronic Notes in Theoretical Computer Science 318 (2015) 91–108.

[14] M. Aldossary, 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 - Volume 1: CLOSER,, INSTICC, SciTePress, 2018, pp. 384–391.

[15] M. Aldossary, I. Alzamil, K. Djemame, Towards virtual machine energy-aware cost prediction in clouds, in: C. Pham, J. Altmann, J. Á. Bañares (Eds.), Economics of Grids, Clouds, Systems, and Services, Springer International Publishing, 2017, pp. 119–131.

[16] X. Zhang, J. Lu, X. Qin, BFEPM: Best fit energy prediction modeling based on CPU utilization, 2013, pp. 41–49.

[17] X. Fan, W.-D. Weber, L. A. Barroso, Power provisioning for a warehouse-sized computer, Vol. 35, 2007, pp. 13–23.

[18] C. Fehling, F. Leymann, R. Retter, W. Schupeck, P. Arbitter, Cloud Computing Patterns, Springer, 2014.

[19] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.

[20] R. N. Calheiros, E. Masoumi, R. Ranjan, R. Buyya, Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications ' QoS, IEEE Transactions on Cloud Computing 3 (4) (2015) 449–458.

[21] G. E. P. B. Cox, D. R., An Analysis of Transformations, Journal of the Royal Statistical Society. Series B (Methodological) 26 (1964) 211–252.

[22] Core, R: A Language and Environment for Statistical Computing, http://www.r-project.org/, Last visited on 03-04-2018.

[23] Stress-ng, Stress-ng, https://www.cyberciti.biz/faq/stress-test-linux-unix-server-with-stress-ng/, Last visited on 03-04-2018.

[24] A. Kostopoulos, A. Dimakis, E. Agiatzidou, Energy-aware Pricing within Cloud Environments, in: 13th International Conference on Economics of Grids, Clouds, Systems and Services (GECON 2016), 2016, pp. 144–159.

[25] OpenNebula, The Simplest Cloud Management Experience, https://opennebula.org/, Last visited on 03-04-2018.

[26] KVM, Kernel-based Virtual Machine, https://www.linux-kvm.org/, Last visited on 03-04-2018.

[27] Watt's-Up, Watts-Up? Plug Load Meters, last visited on 03-04-2018. URL `https://www.powermeterstore.com/P1207/watts_up.php`

[28] Zabbix, The Enterprise-Class Monitoring Solution for Everyone, https://www.zabbix.com/, Last visited on 03-04-2018.

[29] Rackspace, Rackspace, Cloud Servers Pricing and Cloud Server Costs, https://www.rackspace.com/cloud/servers/pricing, Last visited on 03-04-2018.

[30] ElasticHosts, Elastichosts, Pricing - ElasticHosts Linux, Windows VPS Hosting, https://www.elastichosts.co.uk/pricing/, Last visited on 03-04-2018.

[31] VMware, VMware - OnDemand Pricing Calculator, https://vcloud.vmware.com/uk/service-offering/pricing-calculator/on-demand, Last visited on 03-04-2018.

[32] CompareMySolar, Electricity Price Electricity Price per kWh Comparison of Big Six Energy Companies - CompareMySolar.co.uk, http://blog.comparemysolar.co.uk/electricity-price-per-kwh-comparison-of-big-six-energy-companies/, Last visited on 03-04-2018.

[33] R. J. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice, http://otexts.org/fpp/, Last visited on 03-04-2018 (2013).

[34] C. Gu, H. Huang, X. Jia, Power metering for virtual machine in cloud computing-challenges and opportunities, IEEE Access 2 (2014) 1106–1116.

[35] F. Quesnel, H. K. Mehta, J. M. Menaud, Estimating the power consumption of an idle virtual machine, in: 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 2013, pp. 268–275.

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

[37] D. Armstrong, R. Kavanagh, K. Djemame, ASCETiC Project: D2.2.2 Architecture Specification - Version 2, http://www.ascetic.eu/docs/Architecture.pdf, Last visited on 03-04-2018 (2014).

[38] J. Patel, V. Jindal, I. L. Yen, F. Bastani, J. Xu, P. Garraghan, Workload Estimation for Improving Resource Management Decisions in the Cloud, in: Proceedings - 2015 IEEE 12th International Symposium on Autonomous Decentralized Systems, ISADS 2015, 2015, pp. 25–32.

[39] L. Zhang, Y. Zhang, P. Jamshidi, L. Xu, C. Pahl, Service workload patterns for Qos-driven cloud resource management, Journal of Cloud Computing 4 (1) (2015) 1–21.

[40] F. Farahnakian, P. Liljeberg, J. Plosila, LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers, in: Proceedings - 39th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2013, 2013, pp. 357–364.

[41] J. Altmann, M. M. Kashef, Cost model based service placement in federated hybrid clouds, Future Generation Computer Systems 41 (2014) 79–90.

[42] A. Horri, G. Dastghaibyfard, A Novel Cost Based Model for Energy Consumption in Cloud Computing, The Scientific World Journal, Hindawi 2015 (2015) 1–10.

[43] Amazon, Elastic Compute Cloud, last visited on 01-04-2018.
URL `http://aws.amazon.com`

[44] M. Al-Roomi, S. Al-Ebrahim, S. Buqrais, I. Ahman, Cloud Computing Pricing Models: a Survey, International Journal of Grid and Distributed Computing 6 (5).

[45] M. Aldossary, K. Djemame, Energy Consumption-based Pricing Model for Cloud Computing, in: 32nd UK Performance Engineering Workshop, 2016, pp. 16–27.

[46] C. Wang, N. Nasiriani, G. Kesidis, B. Urgaonkar, Q. Wang, L. Chen, A. Gupta, R. Birke, Recouping Energy Costs from Cloud Tenants: Tenant Demand Response Aware Pricing Design, in: ACM 6th International Conference on Future Energy Systems, 2015, pp. 141–150.

[47] A. Dimakis, A. Kostopoulos, E. Agiatzidou, Economic Implications of Energy-Aware Pricing in Clouds, in: 14th International Conference on Economics of Grids, Clouds, Systems and Services (GECON 2017), 2017, pp. 132–144.