Towards an Energy-Aware Cloud Architecture for Smart Grids

Richard Kavanagh[1], Django Armstrong[1] , Karim Djemame[1] , Davide Sommacampagna[2] and Lorenzo Blasi[2]

[1] School of Computing, University of Leeds,
Leeds, LS2 9JT, UK
{R.E.Kavanagh,D.J.Armstrong,K.Djemame}@leeds.ac.uk
http://www.comp.leeds.ac.uk
[2] Hewlett Packard Italiana Srl, Italy.
{Davide.Sommacampagna,Lorenzo.Blasi}@hp.com

Abstract. *Energy consumption in Cloud computing is a significant issue in regards to aspects such as the cost of energy, cooling in the data center and the environmental impact of cloud data centers. Smart grids offers the prospect of dynamic costs for a data center's energy usage. These dynamic costs can be passed on to Cloud users providing incentives for users to moderate their load while also ensuring the Cloud providers are insulated from fluctuations in the cost of energy. The first step towards this is an architecture that focuses on energy monitoring and usage prediction. We provide such an architecture at both the PaaS and IaaS layers, resulting in energy metrics for applications, VMs and physical hosts, which is key to enabling active demand in cloud data centers. This architecture is demonstrated through our initial results utilising a generic use case, providing energy consumption information at the PaaS and IaaS layers. Such monitoring and prediction provides the groundwork for providers passing on energy consumption costs to end users. It is envisaged that the resulting varying price associated with energy consumption can help motivate the formation of methods and tools to support software developers aiming to optimise energy efficiency and minimise the carbon footprint of Cloud applications.*

Keywords: Cloud Computing, Energy Efficiency, Energy, Power, Monitoring, IaaS, PaaS

1 Introduction

Energy efficiency in Cloud computing is fast becoming a primary concern of Cloud providers. Cloud computing is undergoing rapid adoption which is consequently giving rise to a dramatic increase in energy consumption. Data centers consequentially have an ever increasing importance in regards to attempts to save on energy consumption. In addition to this they are important in regards to demand response [2] both due to their high energy usage and the prospect of smoothing their demand through the use of dynamic pricing. Dynamic pricing can therefore be utilised to align a Cloud's power demand to the smart grid.

The first step in creating a pricing mechanism is choosing the correct granularity of product [1] and ensuring the relevant data is available to support the billing process. The fundamental unit of trade remains a virtual machine (VM). However either the price charged for the VM must be able to change during the day or the resource allocation of the VM, in order to cope with the fluctuations in the smart grid's energy market. These changes would be based on both demand for VMs and the current cost of the power consumption of the data center. Thus ensuring end users are motivated to reduce their energy consumption at times when it is most costly.

To this end we present a framework that enables the measurement of the energy efficiency of service deployments in Cloud environments. We discuss this framework and in particular focus on its energy modelling and profiling capabilities at both the Platform as a Service (PaaS) and Infrastructure as a Services (IaaS) layers. The framework is capable of modelling, measuring and reporting on energy efficiency for both billing purposes. These monitoring capabilities offer further advantages in that they support active demand [6], enabling those deploying applications in the Cloud to be made aware of the energy consumption of their applications. This is expected to lead to tooling that can assist developers in understanding and minimising their overall energy consumption.

We demonstrate its capabilities by utilising a 3-Tier web application [20] as a generic cloud use case. We utilise this use case as a means of testing the framework and representing a general workload that might be experienced within a Cloud environment, showing the power utilisation of the distributed application and that of its sub-components. This paper's main contributions are:

- A Cloud architecture that is energy aware.
- support for energy modelling at both the PaaS and IaaS layers, within the architecture.
- a demonstration of recording power consumption via a 3-Tier Web application use case, thus enabling the prospect of billing for energy usage and informing users of the origins of the cost of VMs.

The remaining structure of the paper is as follows: The next section discusses related work. Section 3 discusses the framework's architecture and in particular the energy modelling components at the IaaS and PaaS layers. This is followed by Section 4 that discusses in depth the IaaS and PaaS models that support energy modelling and future predictions of overall energy consumption. The presented architecture and models are then evaluated in Section 5 via a use case in order to demonstrate the effectiveness of the overall architecture, both in the deployment and operation phases. Finally we conclude and present our future work in Section 6.

2 Related Work

The All4Green [2] project focused on Service Level agreements, including supply demand agreements, between energy providers and data centers. These form the basis of providing more flexible tariffs for a data center's energy usage and enabling a demand response mechanism between energy providers and data centers. The architecture proposed in this paper [5] focuses upon energy awareness inside the data center, which is achieved for physical resources, Virtual Machines (VMs) and applications running within the cloud infrastructure. This awareness thus leads towards enabling billing based in part upon energy consumption.

The characterisation of the resources is an important step in regards to accurate energy predictions for software usage. This gives rise to profiling and testing frameworks such as JouleUnit [21] that enable the profiling of hardware systems in order to understand their power consumption profiles. In order to utilise profiling in a more general case monitoring frameworks such as Zabbix [23] or other frameworks [8] may be used. Kwapi [17] is the most closely related monitoring tool to our own work, in that it focuses on power and energy monitoring, however the focus of our framework is upon extending this to both VMs and applications.

Data for resources' power consumption is principally obtained either by direct measurement [9] or inferred via software and physical performance counters [13, 21]. Direct measurement obtains the wall power [13] value via the use of Watt meters [9], providing an aggregation of the current power usage of a physical resource [4]. Performance counters [3, 10, 22] are a non-invasive means of determining energy usage, by utilising performance counters located within the CPU and Operating System. Wall power measurements have the advantage of accuracy but require the specialist physical hardware to be attached into the infrastructure, while the performance counters are indirect measures of power consumption and requires a model to derive an estimate of the energy consumed.

In order to determine VM or host energy usage various frameworks have been developed. The majority of cases use linear models [3,7,13,19] while others lookup table structures [12] have been used or other techniques [22]. In most cases linear models have provided power estimates with a high degree of accuracy for VMs and their underlying resources, usually within 3W of the actual value or within 5% error. Additive models such as [13, 19] utilise load characteristics for each of the major physical components such as CPU, disk and network, each of which is considered separately and summed together. In these cases idle power consumption is treated as an additional model parameter that is simply added to the other load characteristics. There are others that use a bias mechanism [3] or where each parameters importance is learned [22]. The use of performance counters can also differ among models, such as in some cases physical meters are needed during training followed by the use of counters post training [12].

The second concern after profiling a physical host's power consumption is to determine its future energy consumption, which can then be used to guide both the deployment and operation of VMs. Estimating future energy consumption requires an understanding of the VMs workload over time. This can include work such as CPU load prediction in models such as LiRCUP [7] which was aimed at assisting in the maintenance of service level agreements and others [14] that search for workload patterns. Workload prediction has enjoyed a lot of attention with a particular focus on the cloud property of the scaling of resources and the maintenance of QoS parameters [11, 16, 18]. Workload prediction in Clouds has also been seen as a means to plan future workloads so that physical hosts may be switched off when not required [15].

3 Architecture

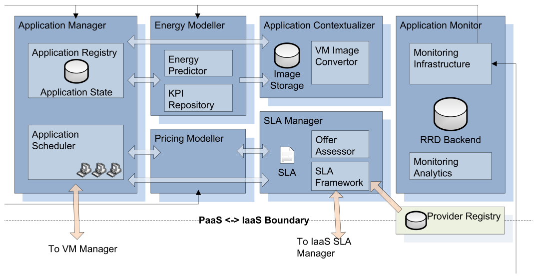


Fig. 1: The PaaS Layer

The architecture follows the standard Cloud deployment model and is hence separated out into the Software as A Service (SaaS), PaaS and IaaS layers. The focus here will be on the PaaS and IaaS layer in which the energy modelling and profiling is performed. Figures 1 and 2 therefore show the high-level interactions of the IaaS and PaaS layer components, in the architecture.

In the SaaS layer (described further in [5]) a collection of components that facilitate the modelling, design and construction of a Cloud applications, has been developed. These components aid in evaluating energy consumption of a Cloud application during its construction and are provided as plug-ins to the Eclipse Integrated Development Environment (IDE). A number of packaging components that facilitate provider agnostic deployment of cloud applications, while also maintaining energy awareness are also provided.

The PaaS layer as shown in Figure 1 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 and tailoring the application to the selected provider's hardware environment. Application level monitoring is also accommodated for here. The core components of this layer are discussed next:

The Application Manager (AM) is the principle component in charge of this layer. It manages the user applications that are described as virtual appliances, formed by a set of VMs that are interconnected between them. It utilises information from the PaaS Energy Modeller and the Pricing Modeller in order to assist in the deployment and operation time running of applications.

The Virtual Machine Contextualizer (VMC) is used to embed software dependencies of a service into a VM image and configure these dependencies at runtime via an infrastructure agnostic contextualization mechanism. This includes the insertion of probes that enable the energy modelling process.

The PaaS Energy Modeller gathers and manages energy related information throughout the Cloud Service lifecycle. It provides an interface to estimate the energy cost of a PaaS KPIs in order to assist in the selection of the appropriate IaaS provider for running an application.

The Application Monitor (APPM) is able to monitor the resources (CPU, memory, network ...) that are being consumed by a given application, by providing historical statistics for host and VM metrics.

The Service Level Agreement Manager is responsible for managing Service Level Agreements (SLAs) at the PaaS level. This requires interacting with the Application Manager, the Pricing Modeller and the IaaS SLA Manager, to establish the terms of any SLA.

The Pricing Modeller provides energy-aware cost estimation and billing information relating to the operation of applications and their VMs.

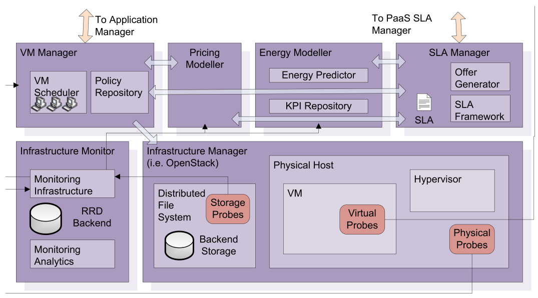


Fig. 2: The IaaS Layer

The IaaS layer as shown in Figure 2 is responsible for the admission, allocation and management of virtual resource. This is achieved through the orchestration of a number of components, that are detailed below:

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. This is achieved with power consumption as a key concern, which is monitored, estimated and optimized using the IaaS Energy Modeller.

The IaaS Energy Modeller has several key features; it is primarily aimed at reporting both host and VM level energy usage data. These values are reported as: a historical log, current values or future predictions. The historical log provides values for energy consumed over a specified time period and the average power, while the current values report power alone. Future predictions are based upon linear regression of CPU utilisation vs power consumption as found via a training phase on a per host basis. The modeller also provides automatic calibration of hosts for training. Predictions utilise either a predefined CPU utilisation level or use a sliding window where the average CPU utilisation over the last N seconds is used as an estimate of future CPU utilisation levels.

The SLA Manager is responsible to manage SLA negotiation requests at IaaS level. This is required in order to allow the PaaS SLA Manager to provide offering for User/SaaS that want to negotiate SLA with the platform.

The Pricing Modeller has the role to provide energy-aware cost estimation and billing information related to the operation of the physical resources that belong to the IaaS provider and are used by specific VMs.

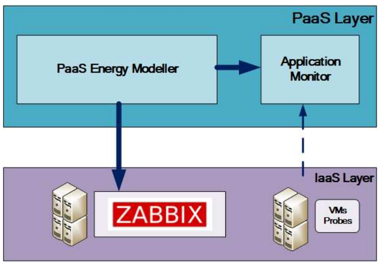
The Infrastructure Manager (IM) manages the physical infrastructure and redirects requests to hardware components. It maintains lists of hardware energy meters, compute nodes, network components and storage devices.

4 Energy Modelling

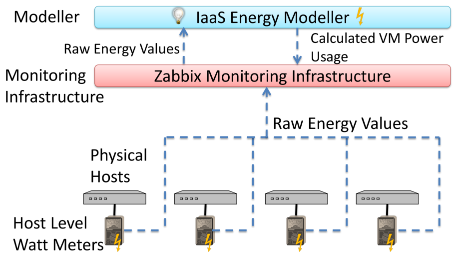
In this section we discuss energy modelling in the two modellers, for applications in the PaaS layer and for VMs and physical resources in the IaaS layer.

4.1 PaaS Energy Modelling

The PaaS Energy Modeller aggregates virtual machines' power measurements generated by the IaaS Layer, into energy and power measurements for applications. It gathers application events, sent by probes to the application monitor, to further support energy profiling at the level of method calls and subcomponents.



(a) PaaS Energy Modeller data sources



(b) Monitoring in the IaaS layer

Fig. 3: Energy Modelling in the PaaS and IaaS layers

Data required by the PaaS Energy Modeller, as illustrated in Figure 3a, is made available by two interfaces. The first provided by Zabbix retrieves VM power measurements generated by the IaaS Energy Modeller. The second provided by the Application Monitor, retrieves events reported by probes installed inside the application's virtual machines. The PaaS Energy Modeller provides the following measurements:

Average Power (W): The average power of an application (1) is the sum of the average power of each one of the n virtual machines (VM_i) where the application is running, which is calculated (2) by averaging all the instant k power measurements collected on VM_i during that application's lifetime.

$$\bar{P}_{Application} = \sum_{i=1}^{n} \bar{P}_{VM_i} \tag{1}$$

$$\bar{P}_{VM_i} = \frac{1}{k} \sum_{j=1}^{k} \bar{P}_j^i \tag{2}$$

Energy Consumed (Wh): The total energy consumed by the application (3) is the sum of its virtual machines' consumption, which is calculated (4) from the sum of the energy between each pair of power measurements in two consecutive instants, say t_a and t_b, by applying the trapezoidal rule (5).

$$E_{Application} = \sum_{i=1}^{n} E_{VM_i} \tag{3}$$

$$E_{VM_i} = \sum E^i_{t_b t_a} \tag{4}$$

$$E^i_{t_b t_a} = (t_b - t_a)\frac{P^i(t_a) + P^i(t_b)}{2} \tag{5}$$

The two measurements of power and energy described above are about the whole application; for finer analysis, the PaaS EM, can use events' timestamps to restrict the power calculation to the time interval when the event occurred.

Event data: Every time an event occurs its data is collected by probes; in this way the Energy Modeller knows when such event started, terminated and on which VM it occurred. As shown in Figure 4, probe data provides the start and end time stamps of the event ("starttime" and "endtime"); each event also reports the identifier of its type ("eventType"), which allows identifying different types of events for the same application to be analyzed separately.

Average Power per Event Type (W): The PaaS Energy Modeller uses events data to collect measurements of power from the virtual machine where each event occurred, within a specified time period; then it calculates the average of all measurements for each event and finally returns the average power calculated on all events of the same type.

Energy Consumed per Event Type (Wh): The PaaS Energy Modeller collects events data and computes the discrete integral, using (5) for each pair

```
<event>
      <applicationID>HMMERpfam</applicationID>
      <eventType>eventA</eventType><instanceId>45</instanceId>
      <nodeId>4351a79d-75ee-42fd-b9e6-b963a10b1787</nodeId>
      <starttime>1411468096408</starttime>
      <endtime>1411468096930</endtime>
</event>
```


Fig. 4: Event Data Example

of measurements of the power, to obtain the total energy consumption. Finally it sums the single events' consumption for each event of the same type to get the total energy consumption for that event type. In order to provide a better approximation of the energy consumed by each event, the PaaS Energy Modeller interpolates power values from the available samples. This approximation is required for events lasting a short period of time.

4.2 IaaS Energy Modelling

The IaaS Energy Modeller has two main roles, the first is at deployment time when the VM Manager utilises power consumption predictions of VMs. The second is at operation time when the VMs are monitored. In both cases the IaaS Energy Modeller is required to attribute power consumption to existing VMs or VMs that are scheduled to be deployed.

The energy monitoring in the IaaS Layer is shown in Figure 3b. At the lowest level the monitoring utilises Watt meters [9] that are attached to the physical host machines. The data from these meters is published in Zabbix [23]. The values for host power consumption is then read by the IaaS Energy Modeller. The Energy Modeller's main role is to assign energy consumption values to a VM from the values obtained at host level. This is needed because energy consumption associated with VMs is not a directly measurable concept. Rules therefore establish how the host energy consumption is assigned to VMs. The host energy consumption can be fractioned out in one of several ways, within the Energy Modeller which is detailed below:

CPU Utilisation Only: This uses CPU utilisation data for each VM and assigns the energy usage by the ratio produced by the utilisation data. (Available for: Historic, Current, Predictions). This is described in the Equation 6 where VM_P_x is the named VM's power consumption, $Host\_P$ is the measured host power consumption. VM_Util_x is the named VMs CPU utilisation, VM_Count is the count of VMs on the host machine. VM_Util_y is the CPU utilisation of a member of the set of VMs on the named host.

$$VM\_P_x = Host\_P \times \frac{VM\_Util_x}{\sum_{y=1}^{VM\_Count} VM\_Util_y} \quad (6)$$

CPU Utilisation and Idle Energy Usage: Idle energy consumption of a host can also be considered. Using training data the idle energy of a host is calculated. This is evenly distributed among the VMs that are running upon the host machine. The remaining energy is then allocated in a similar fashion to the CPU Utilisation only mechanism. (Available for: Historic, Current, Predictions). This is described in Equation 7 where $Host\_Idle$ is the host's measured idle power consumption. This provides an advantage over the first method in that a VM is more appropriately allocated power consumption values and prevents it from using no power while it is instantiated.

$$VM\_P_x = Host\_Idle + (Host_P - Host\_Idle) \times \frac{VM\_Util_x}{\sum_{y=1}^{VM\_Count} VM\_Util_y} \quad (7)$$

Evenly Shared: In the case of predictions CPU utilisation is not always clearly estimable, thus energy consumption can be evenly fractioned amongst VMs that are on the host machine. The default for predictions is to share out power consumption evenly as per Equation 8. A slight variation also exists which counts the CPU cores allocated to each of the VMs and allocating based upon this count (Equation 9). Equations 8 and 9 describe this even sharing rules where $Host\_Predicted$ is the amount of power that the host on which the named VM resides is estimated to utilise and VM_VCPU_x is the amount of virtual CPUs allocated to the named VM while VM_VCPU_y is the amount of virtual CPUs allocated to a VMs on the named host.

$$VM\_P_x = Host\_Predicted \times \frac{1}{VM\_Count} \quad (8)$$

$$VM\_P_x = Host\_Predicted \times \frac{VM\_VCPU_x}{\sum_{y=1}^{VM\_Count} VM\_VCPU_y} \quad (9)$$

The default method chosen on the IaaS Energy Modeller is option 2 for current and historic values and 3 for predictions. Once the Energy Modeller has assigned energy values to a given VM it then writes these values to disk, which are again via Zabbix_sender reported back to the monitoring infrastructure, thus providing VM level power consumption values to the PaaS layer.

5 Evaluation

In this section we present an evaluation of our framework. The overall aim of the experimentation is demonstrate the energy monitoring capabilities of the proposed architecture and show how it can ensure customers are aware of their energy usage for Cloud data centers, thus achieving the awareness aspect of active demand [6]. This is achieved by running a 3-Tier Web application [20], which has been chosen as an example of a typical Cloud/Web based application. In running this experiment we demonstrate the usage of the energy modellers at the PaaS and IaaS layers, which is the first step towards billing end users for a VM's energy consumption. The objectives of the experiment will be to find at the PaaS level the overall application's energy usage and at the IaaS level the energy usage per VM. This will be done under varying workload conditions. This will therefore highlight how the framework can both advise end users of their application's energy consumption which enables transparency during billing and provide a means for recording the energy consumption of an individual user. In the next section we discuss the experimental setup, followed by a discussion of our findings in Section 5.2.

5.1 Experimental Setup

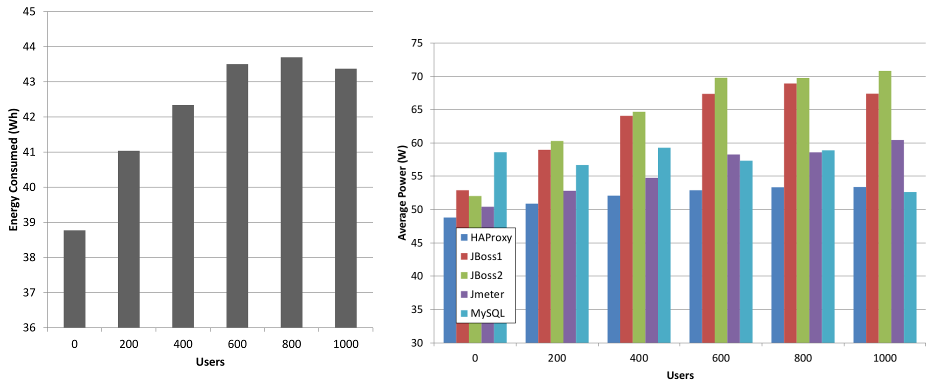
The 3-Tier web application is the RichFaces Photo Album as provided by `jboss.org`. It has been configured to run on five VMs one containing haproxy and one containing a MySQL database. JMeter was utilised on a third to induce workload and simulate the users during the experimentation. The final two VMs ran instances of JBoss, to execute the workload that was induced. The VMs were deployed using OpenStack.

The experiment was run with 4 physical hosts available. Each host is equipped with two quad-core processors with 2.66 GHz, 32 GB of RAM and a 750 GB local hard disk. Each host is connected to two different networks transferring one Gbit/s each synchronously. The first is dedicated for infrastructure management as well as regular data exchange between the physical hosts. The second is for the storage area network usage only where storage nodes are accessible through a distributed file system. Monitoring data including the wall power is made available through the 1st network. Each energy-meter can measure voltage, current and power consumption. The energy meter's used are *Gembird EnerGenie Energy Meters* [9]. These energy measurements are reported once per second, through Zabbix 2.2. One host is dedicated to collecting the measurements and can share the aggregated information with the monitoring components and another hosts is dedicated towards acting as the main storage management system.

5.2 Results & Discussion

In the first experiment the 5 VMs as previously described will be run, with varying amounts of users trying to access the system. The users will vary from 0 to 1000 in increments of 200. The VMs will be started providing a deployment phase for each run, followed by a period of undergoing load from users. Each run will be repeated 5 times in order to ensure validity of the results. The PaaS level metric of the application's overall energy consumed will then be presented along with IaaS layer metric of the energy consumption per each VM.

Figure 5a shows the energy consumed by the Photo Album, for a range of different users. We can see that the amount of energy consumed for each run increases with the amount of users, up until a plateaus at 600 users due to the workload saturating the VMs. Figure 5b shows the average power consumed by each VM in the Photo Album, for a range of different users. In a similar fashion to Figure 5a the plateau can be seen. It can also be observed the relative energy usage in each of the VMs, with substantial increases in the JBoss instances power



(a) The total energy consumed by the application

(b) The per VM average power consumption

Fig. 5: Energy consumption in each layer

consumption and more limited increases from the JMeter instance. The HAProxy and MySQL instances have relatively modest increases in power consumption. This can be seen as a clear demonstration of how different instance types can be identified as more energy consuming than others and of their relative power consumption.

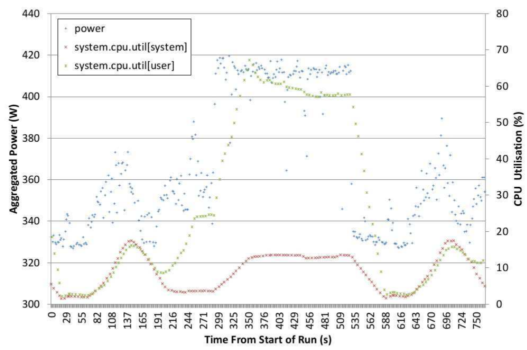


Fig. 6: Trace of the power consumed during the life of the VMs

Figure 6 shows a trace of the CPU utilisation and aggregated power consumption of VMs during a single experimental run, for 1000 users. The deployment, running and undeployment phases are shown as the spikes on the graph. This illustrates the energy awareness, which can be used to choose providers and optimise placement of VMs as well as by developers to compare deployment strategies and iteratively reduce energy consumption of applications.

6 Conclusion and Future Work

We have demonstrated how the proposed framework can measure an applications overall energy consumption and how this can be broken down on a per VMs basis. This research can be leverage by future smart grids from two perspectives that help manage demand. Firstly, it enables user awareness of energy consumption and secondly, facilitates billing based on this information. This ability to measure power consumption also gives developers the opportunity to examine how their applications perform in terms of energy consumption on the Cloud. In future work the PaaS Energy Modeller will be equipped with a statistical model to be trained before application execution. This model will provide short term forecasting capabilities to support real time monitoring in order to implement early detection of SLA violations and therefore proactive enforcement. The IaaS layer will be expanded upon to include greater profiling of VM types in workload prediction and improved profiling of resources. The pricing model inside the framework will be expanded beyond static pricing and will utilise the estimates of power and energy consumption in order to set the price for VMs.

7 Acknowledgements

This work has received support through the EU 7th Framework Programme for research, technological development and demonstration under Grant Agreement no 610874 as part of the ASCETiC project.

References

1. Altmann, J., Neumann, D., Fahringer, T., Courcoubetis, C., Stamoulis, G., Dramitinos, M., Rayna, T., Risch, M., Bannink, C.: GridEcon: A Market Place for Computing Resources. In: Grid Economics and Business Models, vol. 5206, pp. 185–196. Springer Berlin / Heidelberg (2008)
2. Basmadjian, R., Lovasz, G., Beck, M., de Meer, H., Hesselbach-Serra, X., Botero, J.F., Klingert, S., Perez Ortega, M., Lopez, J.C., Stam, A., van Krevelen, R., di Girolamo, M.: A Generic Architecture for Demand Response: The ALL4Green Approach (2013)
3. Bohra, A.E.H., Chaudhary, V.: VMeter: Power modelling for virtualized clouds. In: Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on. pp. 1–8 (Apr 2010)
4. Castañé, G.G., Núñez, A., Llopis, P., Carretero, J.: E-mc2: A formal framework for energy modelling in cloud computing. Simulation Modelling Practice and Theory 39(0), 56–75 (2013)
5. Djemame, K., Armstrong, D., Kavanagh, R., Ferrer, A.J., Perez, D.G., Antona, D., Deprez, J.C., Ponsard, C., Ortiz, D., Macias, M., Guitart, J., Lordan, F., Ejarque, J., Sirvent, R., Badia, R., Kammer, M., Kao, O., Agiatzidou, E., Dimakis, A., Courcoubetis, C., Blasi, L.: Energy efficiency embedded service lifecycle: Towards an energy efficient cloud computing architecture. In: CEUR Workshop Proceedings. vol. 1203, pp. 1–6. CEUR-WS (2014)

6. European Technology Platform on Smart Grids: Consolidate View of ETP SG on Smart Grids (European Technology Platform on Smart Grids). Tech. rep. (2015)
7. Farahnakian, F., Liljeberg, P., Plosila, J.: LiRCUP: Linear Regression Based CPU Usage Prediction Algorithm for Live Migration of Virtual Machines in Data Centers. In: Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on. pp. 357–364 (Sep 2013)
8. Fatema, K., Emeakaroha, V.C., Healy, P.D., Morrison, J.P., Lynn, T.: A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives. Journal of Parallel and Distributed Computing 74(10), 2918–2933 (2014)
9. GEMBIRD Deutschland GmbH: EGM-PWM-LAN data sheet. http://gmb.nl/Repository/6736/EGM-PWM-LAN_manual—7f3db9f9-65f1-4508-a986-90915709e544.pdf (2013)
10. Hähnel, M., Döbel, B., Völp, M., Härtig, H.: Measuring Energy Consumption for Short Code Paths Using RAPL. SIGMETRICS Perform. Eval. Rev. 40(3), 13–17 (Jan 2012)
11. Islam, S., Keung, J., Lee, K., Liu, A.: Empirical prediction models for adaptive resource provisioning in the cloud. Future Generation Computer Systems 28(1), 155–162 (Jan 2012)
12. Jiang, Z., Lu, C., Cai, Y.: VPower: Metering power consumption of VM. 2013 IEEE 4th International Conference on Software Engineering and Service Science pp. 483–486 (May 2013)
13. Kansal, A., Zhao, F., Liu, J., Kothari, N., Bhattacharya, A.A.: Virtual Machine Power Metering and Provisioning. In: Proceedings of the 1st ACM Symposium on Cloud Computing. pp. 39–50. SoCC '10, ACM, New York, NY, USA (2010)
14. Khan, A., Yan, X., Tao, S., Anerousis, N.: Workload characterization and prediction in the cloud: A multiple time series approach (2012)
15. Prevost, J.J., Nagothu, K., Jamshidi, M., Kelley, B.: Optimal calculation overhead for energy efficient cloud workload prediction (2014)
16. Quiroz, A., Kim, H., Parashar, M., Gnanasambandam, N., Sharma, N.: Towards autonomic workload provisioning for enterprise Grids and clouds (2009)
17. Rossigneux, F., Lefevre, L., Gelas, J.P., Dias De Assuncao, M.: A Generic and Extensible Framework for Monitoring Energy Consumption of OpenStack Clouds. In: The 4th IEEE International Conference on Sustainable Computing and Communications (Sustaincom 2014). Sydney, Australia (Dec 2014)
18. Roy, N., Dubey, A., Gokhale, A.: Efficient Autoscaling in the Cloud Using Predictive Models for Workload Forecasting (2011)
19. Smith, J.W., Khajeh-Hosseini, A., Ward, J.S., Sommerville, I.: CloudMonitor: Profiling Power Usage. In: Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. pp. 947–948 (Jun 2012)
20. Tsebro, A., Mukhina, S., Galkin, G., Sorokin, M.: RichFaces Photo Album Application Guide (2009), `http://docs.jboss.org/richfaces/latest\_3\_3\_X/en/realworld/html\_single/`
21. Wilke, C., Götz, S., Richly, S.: JouleUnit: A Generic Framework for Software Energy Profiling and Testing. In: Proceedings of the 2013 Workshop on Green in/by Software Engineering. pp. 9–14. GIBSE '13, ACM, New York, NY, USA (2013)
22. Yang, H., Zhao, Q., Luan, Z., Qian, D.: iMeter: An integrated {VM} power model based on performance profiling. Future Generation Computer Systems 36(0), 267–286 (2014)
23. ZABBIX SIA: Homepage of Zabbix:: An Enterprise-Class Open Source Distributed Monitoring Solution (2014), `http://www.zabbix.com/`