Measuring Energy Consumption for Web Service Product Configuration

I Made Murwantara, Behzad Bordbar, Leandro L. Minku
School of Computer Science
University of Birmingham
Birmingham, B15 2TT, UK
{ixm249, b.bordbar, l.l.minku}@cs.bham.ac.uk

ABSTRACT

Because of the economies of scale that Cloud provides, there is great interest in hosting web services on the Cloud. Web services are created from components such as Database Management Systems and HTTP servers. There is a wide variety of components that can be used to configure a web service. The choice of components influences the performance and energy consumption. Most current research in the web service technologies focuses on system performance, and only small number of researchers give attention to energy consumption. In this paper, we propose a method to select the web service configurations which reduce energy consumption. Our method has capabilities to manage feature configuration and predict energy consumption of web service systems. To validate, we developed a technique to measure energy consumption of several web service configurations running in a Virtualized environment. Our approach allows Cloud companies to provide choices of web service technology that consumes less energy.

Categories and Subject Descriptors

D.2.13 [**Software Engineering**]: Reusable Software-Reuse models

General Terms

Measurement, Design

Keywords

Energy Aware, Software Product Line, Web System, Machine Learning

1. INTRODUCTION

In 2013, power consumption of data centres reached about 10% of the world power consumption [5], where most of data centres host Cloud computing systems. It is predicted that in the year 2020 the power consumption may increase into 100 GigaWatt [17]. A Cloud system is a massive shared infrastructure that may consist of thousands of servers. Considering the ongoing increase in the cost of energy, efficient energy management of application within Cloud systems will reduce significant expenses.

A large number of applications in Cloud systems are web-based applications. These applications can be configured from a typical combination of HTTP server, web application and database system. When configuring the web-based system, different combinations of components give distinct energy consumption. If we find the combination that consumes less, because of the scale of the economy, save would be a lot.

To improve energy efficiency of a web service, we need a framework to capture combinations of components. We use Software Product Line Engineering (SPLE) [6] notation to model the selection of combinations of web service components. So energy consumption of a combination of a web service can be modeled based on a given workload. For example, we can measure the energy consumption of a combination of an HTTP server, a Web application and a Database system given a numbers of incoming requests. This method helps us to find suitable configurations with their prediction of energy usages. Furthermore, SPLE method is already proven to handle complex configuration with different types of attributes [15]; some components can be combined together and some combinations of components are not composable. SPLE notation allows specifying permissible components.

The number of combinations of web service technologies and workload patterns can be large. Therefore, we use machine learning to learn how to predict the energy consumption of different web service without having to measure the energy consumption of all possible combinations and workloads. Our method can be used by Cloud providers and companies affiliated to Cloud system to find the best combination and encourage application developers and users to use configurations that consume less energy. To show the feasibility of our approach, we build a case study using Wordpress [3], an open source blog system that run on PHP-based web service. And to deal with Virtualized environment, we use our energy measurement method [12] that uses software power meter to measure energy usage with a given workload.

The remaining of this paper is as follows. In Section II, we present essential background material on the existing power measurement techniques, software measurement of energy, workload tools, Software Product Line Engineering and Machine Learning. Subsequently, in Section III, we describe the

iiWAS2014, 4-6 December 2014, Hanoi, Vietnam.

http://dx.doi.org/10.1145/2684200.2684314 .

contribution of our research. After that, in Section IV, we shall explain capturing of configurations via SPL and measuring energy consumed by a given configuration, and we will discuss our measurement of energy consumption on several combinations of PHP-based web service in Section V. In Section VI, we will explain the outline of using machine learning to predict power consumption. Then, Section VII presents and compares related work. Section VIII concludes this paper.

2. BACKGROUND

In this section, we shall present background material used in the rest of the paper.

2.1 Measurement of Energy Consumption via Software and Workload Tools

There are numbers of software applications for measuring power consumption in a computer system [11, 10]. Some of these software products use battery drainage to measure how much energy is consumed.

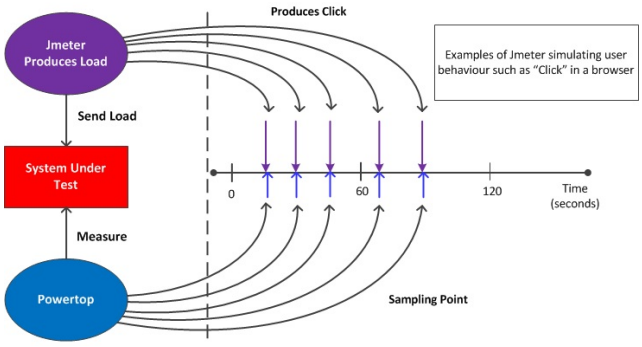

Figure 1: Workload and Measurement [12]

To measure energy consumption of any given application, suitable workload must be provided to simulate its usage. As depicted in Figure 1, we use Jmeter [1] to generate a workload into a system under test and using Powertop to measure how much energy is consumed under the given load.

2.2 PHP Web Service

Most existing systems which our daily activities rely on are web based. As a result, we focus our research and evaluation on web-based applications. Among the key components of a web-based system are the web services. In general, 'web services are client and server applications that communicate over the World Wide Web (WWW) via HyperText Transfer Protocol (HTTP)' [13]. Most web-based IT systems include HTTP server, web application and Database system. In this paper, we use PHP: Hypertext Pre-processor [16], the script interpreter for dynamic web development, is our measurement object.

2.3 Software Product Line Engineering (SPLE)

We need a method to represent configurations of the systems, which we will their energy consumption measured. A Software Product Line (SPL) is 'a set of software-intensive systems sharing a common, managed set of features, and that are developed in a disciplined fashion using a common set of core assets' [6]. SPL is particularly relevant as it

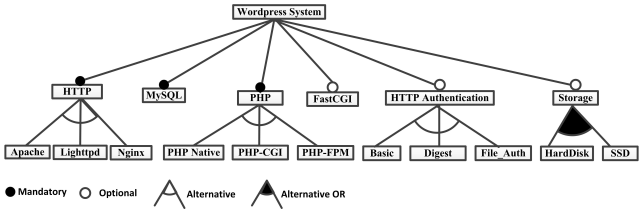

Figure 2: Feature Model of Wordpress Web Technology

allows capturing configuration involving numerous variants for each part. In SPL development, Feature Model (FM) is a well-known modelling technique for representing all of the features in the Software Product Line [7]. A feature is the representation of a functionality or a product included in an SPL. A Feature Model defines the commonality and variability of concept instances and dependencies between features [7].

We build a feature model, as seen in Figure 2, for a typical web service such as PHP web service. A variability feature such as HTTP has sub-features 'Apache', 'Nginx' and 'Lighttpd'. In the feature HTTP, the arc symbol between variant features represents alternative options where one of sub-features must be selected. In feature 'Storage' represents Alternative OR option where at least one of sub-features must be selected. Mandatory and Optional represent required and optional features to include in a configuration.

2.4 Machine Learning

As explained in section 1, a method to predict the CPU power consumption given a workload and configuration is necessary. Building a model to make such predictions can be viewed as a machine learning regression problem, i.e., the problem of learning how to predict a numeric target variable (CPU power) given a set of input variables (workload and configuration). Machine learning methods build predictive models based on a training set composed of examples of input values and their corresponding target outputs. Several different machine learning methods exist for regression problems [4]. In this paper, we will investigate the following popular methods: linear regression, Multilayer Perceptrons (MLPs), Regression Trees (RTs), Bagging ensembles of RTs (Bagging+RTs) and Bagging ensembles of MLPs (Bagging+MLPs).

3. CONTRIBUTION

We have our measurement method [12], which can capture energy consumption of individual processes that are running in Cloud system and Virtualized environment that have their workload from a workload tool.

The contribution of this paper is twofold. The first contribution is a modelling technique using SPL approach to analyse consumption of energy of the web service products. As a result, a web technology developer can find out the combination of web service components that consumes less energy from a number of given software products with comparable functionalities.

As the number of combination and range of workloads is high, it is not efficient to measure all possible combinations. The second contribution is using Machine Learning to pre-

dict the energy consumption of SPL combinations.

In the next section, we will show how the generated workload is used to conduct the measurement of energy usages.

4. ENERGY VALUE FOR SPL CONFIGURATION

In section 2, we explained the outline of our method for measuring energy consumption for a given configuration of products and any given load. There are two tasks involved, firstly, identifying what configurations are permissible. For example, feature 'PHP-Native' requires feature 'Apache' and feature 'Nginx' excludes feature 'PHP-Native'. So, when feature 'PHP-Native' is chosen with feature 'Nginx', it will raise model fault during design stage. Secondly, there is a wide range of configurations. For example, a simple combination of PHP-based web service that includes HTTP servers, PHP technologies and Database systems. Even for this simple example, if we take into account the possible workloads, the number of possible combinations can easily reach hundreds. It is impractical to measure the energy consumed under all possible combinations. We use machine learning to predict energy consumption of the features combinations.

4.1 Using SPL to Identify Configuration

The feature model is proposed to deal with complexity and modeling of an SPL configuration. As we can see in Table 3, Apache server can be configured by three PHP variants that are PHP-Native, PHP-CGI and PHP-FPM. However, Lighttpd servers can only be configured with two of the variants, PHP-CGI and PHP-FPM. The knowledge is missing from the diagram of Figure 2. SPL specification rely on constraints to capture such information. For example, a propositional logic formula (Lighttpd $\wedge\neg$PHP-Native) will enforce that such dependency cannot be configured.

4.2 Capturing dependencies between features that influence energy usage

In order to measure energy consumption, Software Product Line must consider the effect of choosing different features. Combination of possible features can be extracted from functionality that captures their dependencies in a product line.

In the PHP-based web service, the configuration of a product line may have goals on its design. For example, a website system built to handle intensive transactions, such as web for social media, needs to respond fast and reliaby upon users request. As consequence, the design of a web system must be reliable and good in handling high number of requests in a short time.

In our approach, a combination of features is associated with energy consumption. For example, combinations of features 'Apache', 'PHP-CGI' and 'MySQL' will have different consumption of energy compared to the combination of features 'Apache', 'PHP-Native' and 'MySQL'. We group the combination of features to categorize the consumption of energy, where each Product Configuration characterizes the web technology. For example, 'PHP-Native' as a web service component only works with 'Apache' HTTP server. In addition, three options to use 'FastCGI' with Process Manager that are features 'Apache', 'Lighttpd' and 'Nginx'.

Table 1: Energy Consumption of HTTP, Database and CPU in the PHP-based Web System

Configuration	Min(Watt)	Max(Watt)
Apache+PHP-CGI+MySQL	http: 0.018 db: 0.097 cpu: 5.820	http: 0.301 db: 5.820 cpu: 33.9
Apache+PHP-FPM+MySQL	http: 0.014 db: 0.040 cpu: 5.653	http: 0.308 db: 2.289 cpu: 31.38
Apache+PHP-Native+MySQL	http: 0.360 db: 0.077 cpu: 5.869	http: 16.344 db: 2.19 cpu: 30.96
Lighttpd+PHP-CGI+MySQL	http: 0.005 db: 0.065 cpu: 5.214	http: 0.201 db: 2.262 cpu: 29.94
Lighttpd+PHP-FPM+MySQL	http: 0.010 db: 0.107 cpu: 5.027	http: 0.210 db: 2.258 cpu: 31.34
Nginx+PHP-FPM+MySQL	http: 0.015 db: 0.110 cpu: 5.477	cpu: 0.204 db: 2.236 cpu: 36.66

5. EXAMPLE OF MEASURING ENERGY CONSUMPTION FOR A GIVEN CONFIGURATION

In this section, we will discuss on the measurement results of energy consumption of variant features of PHP-based web system. Our method can be easily adapted to measuring other platforms such as Cloud system, Mobile system and TabletPC. All virtual environments in these measurements use similar hardware configuration.

Our method can show the amount of energy consumed by each individual process within a virtual environment such as KVM-based virtual machine in Laptop. We use the software for sampling over a few periods of time-sequence and to calculate average energy consumption. We also change the load to measure the amount of energy under different loads. We have divided the experiment into blocks of 10 seconds for 100 increasing stage. In the first 10 seconds, we produce loads of one user per second, and in the next ten seconds period we produce the load of two users per seconds and so on. This results in 1-100 users per period of 10 seconds.

We can compare the energy consumption between different combinations of PHP web service components, as explained in the next subsection.

5.1 Power Consumption in Different Combination

Measurement of power consumption of the web system in several HTTP servers such as Apache, Nginx and Lighttpd is comparable because it uses the same PHP web technology.

We measure the consumption of energy of several combinations of PHP-based web systems. We use the Wordpress [3] system, a well-known open source blog system, running on top of Virtualized system with virtual machine specification as follows, Linux Ubuntu 13 operating system, 1 CPU core and 1 Giga Byte memory. The measurement runs in several combinations of HTTP servers and variant PHP technology connected to a MySQL database system.

The measurement results from all configurations, as de-

picted in Table 1, show that the Apache HTTP server consumed the highest amount of energy when the web system is configured using PHP-Native web technology. In a database system, MySQL consumed highest energy when it combined Apache HTTP server with PHP-CGI web technology where the other configuration did not influence significantly. For CPU power expenditure, the configuration of Nginx HTTP server with PHP-FPM consumed the highest amount of energy when there were 100 users accessing the system within 10 seconds.

6. USING MACHINE LEARNING FOR CHOOSING CONFIGURATION

In a web system, software components have configurations such as Apache HTTP server with PHP. Each configuration of the web systems may vary on their process and thus consume a different amount of energy. Using Software Product Line Engineering can help on the decision and reuse of the software components with configurations that use less energy. However, measuring the energy consumed by each possible configuration and workload is impractical, due to the very high number of possible combinations.

Feature modelling helps to manage the logical combination of features. Features dependency as a way to manage user requirements rules the SPL configuration. As a result, we check the dependency rules of features before checking possibility of combination of features from the measurement results. Further, we use Machine Learning to predict individual component and configuration energy usage with varied workloads. This can help a software product line developer to decide which configuration to use for a given workload, so as to minimise energy consumption.

The Software Product Line that concerns with energy expenditure for PHP-based web system helps the Cloud companies to estimate the energy consumption that will run on their infrastructure. Another benefit is that the Cloud companies can offer the configuration of PHP-based web system that run efficiently in their infrastructure. Accordingly, the cost of energy can be reduced and performance of the system can be increased.

6.1 CPU Power Prediction

As explained in section 3, we use machine learning for predicting the CPU power consumption of different workloads and configurations. When applying machine learning for a given problem for the first time, two questions should be asked: [Q1] Can we consider the performance of the methods acceptable for the problem in hands? [Q2] What machine learning method is best to use for this problem?

Different machine learning methods have different advantages and disadvantages. These can be beneficial or detrimental to the problem investigated in this paper. So, this section presents a comparative study of the performance of models created by five different popular methods, aiming at answering the questions above. Section 6.1.1 reports the setup of the experiments and section 6.1.2 reports the analysis done to answer the questions above.

6.1.1 Experimental Setup

WEKA's implementation of linear regression, MLP, RT (REPTree), Bagging+MLPs and Bagging+RTs was used with its default parameters in the experiments [9]. Using the default parameters is a reasonable choice for a first analysis, as practitioners would frequently leave parameters untuned unless they are experts in machine learning.

The experiments were based on ten times ten fold cross-validation using a data set created by simulating users accessing a web service. One input variable is used to describe the workload and eight binary variables a to h are used to describe the configuration as shown in table 3. The target variable is the CPU power.

The performance measures used in this study were Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Median Magnitude of the Relative Error (MdMRE):

- $MAE = \frac{1}{T}\sum_{i=1}^{T}|\hat{y}_i - y_i|$;
- $RMSE = \sqrt{\frac{\sum_{i=1}^{T}(\hat{y}_i - y_i)^2}{T}}$; and
- $MdMRE = \text{Median } \{MRE_i | 1 \leq i \leq T\}$, where $MRE_i = 100 \cdot \frac{|\hat{y}_i - y_i|}{y_i}$,

where $\hat{y}_i$ is the predicted CPU power, y_i is the actual CPU power and T is the number of examples used for testing. MAE is a measure recommended for being unbiased towards under and overestimations [14]. RMSE is a popular measure in the machine learning community which emphasizes large errors more. MdMRE can be biased towards underestimations, i.e., it may favour methods that make underestimations [14]. So, MAE and RMSE are more appropriate for comparing methods, whereas MdMRE was included only to give a general idea of the performance in terms of percentage of the actual CPU power levels.

Table 2: Average performance (+- standard deviation) of machine learning methods for predicting CPU power. The MAE and RMSE of all methods was statistically significant different from Bagging+RT's based on Wilcoxon Sign-Rank Test with Holm-Bonferroni corrections at the overall level of significance of 0.05.

Method	MAE	RMSE	MdMRE
Linear regression	0.79+-0.22	1.74+-1.01	2.43+-0.52
MLP	1.17+-0.56	2.07+-1.03	4.41+-2.81
RT	0.72+-0.18	1.28+-0.60	2.64+-0.43
Bagging+MLP	0.89+-0.25	1.86+-1.00	2.66+-0.85
Bagging+RT	0.61+-0.14	1.15+-0.52	2.22+-0.38

6.1.2 Experimental Results

Table 2 shows the results of the average test performance. Given that the average and standard deviation of the target CPU power levels in the full data set were 18.56 and 7.33, we can consider the average MAE and RMSE to be generally small for all methods. This is further confirmed by the very low MdMRE. For instance, the MdMRE for Bagging+RT was only 2.22% of the actual CPU power levels. This answers Q1, showing that the methods investigated in this study present good performance for the problem of predicting CPU power. It is worth noting that we used the default parameters of the methods for these experiments. If the parameters are tuned, even better results may be obtained.

The MAE and RMSE of each method was compared against the highest ranked approach (Bagging+RT) based on Wilcoxon

Table 3: Configuration variables in the data set.

a	b	c	d	e	f	g	h	Configuration
0	0	1	1	0	0	0	1	Apache + PHP-CGI + MySQL
0	0	1	0	1	0	0	1	Apache + PHP-FPM + MySQL
0	0	1	0	0	1	0	1	Apache + PHP-Native + MySQL
1	0	0	1	0	0	0	1	Lighttpd + PHP-CGI + MySQL
1	0	0	0	1	0	0	1	Lighttpd + PHP-FPM + MySQL
0	1	0	0	1	0	0	1	Nginx + PHP-FPM + MySQL

Sign-Rank Test with Holm-Bonferroni corrections at the overall level of significance of 0.05. All p-values were smaller than $2.5 \cdot 10^{-5}$. Therefore, we can conclude that Bagging+RT obtained significantly better MAE and RMSE than the other methods. It is thus recommended over the other methods for predicting CPU power, if the main aim is to achieve low prediction error. This answers Q2.

It is worth noting that the models generated by methods such as linear regression and RT can be visualised, so that one can understand the mapping of input to output variables provided by these algorithms. Due to space limitations, we leave that as future work.

7. RELATED WORK

Dougherty [8] develops a model-driven for auto-scaling Cloud computing infrastructure, where autonomous management of virtual machine can optimize the energy consumption. In our approach, a configuration of features is associated to the energy measurement results. We measure energy consumption on top of a Virtualized environment and using machine learning to predict energy consumption of large product configurations.

Bartalos [2] develops a method to predict energy consumption of a web service using an aggregate linear instantaneous power model, and estimates power consumption using a synthetic web service model. In our approach, we use diverse workload to measure energy consumption of individual processes within different web service configuration, and using Regression Tree of machine learning method to predict the energy consumption.

8. CONCLUSION

In this paper, we used Software Product Line to configure a web service by providing choices of combinations of web services technologies that consumes less energy. We show that it is possible to measure energy consumed by individual processes in a Virtualized environment. As the number of possible configurations can be relatively large, we use machine learning technique to predict the consumption of energy of a particular combination of web service components. When more than one component can perform a given functionality, this technique will allow the users to choose configurations of components, which consumes the least amount of energy.

9. ACKNOWLEDGMENTS

I Made Murwantara was supported by Indonesia Higher Education (DIKTI) project No. BLN120200-2012. Leandro Minku was supported by EPSRC Grant No. EP/J017515/1.

10. REFERENCES

[1] Apache. Apache jmeter. http://jmeter.apache.org. Accessed: 04/01/2014.

[2] P. Bartalos and M. B. Blake. Green web services: Modeling and estimating power consumption of web services. In *ICWS*, pages 178–185, 2012.

[3] M. Beck and J. Beck. *WordPress: Visual QuickStart Guide*. Pearson Education, 2013.

[4] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, USA, 2006.

[5] J. Clark. It electricity use worse than you thought. http://www.theregister.co.uk/2013/08/16/. Accessed: 02/07/2014.

[6] P. Clements and L. Northrop. *Software Product Lines: Practices and Patterns*. The SEI Series in Software Engineering. Prentice Hall, 2002.

[7] K. Czarnecki and U. W. Eisenecker. *Generative programming: methods, tools, and applications*. ACM Press, New York, NY, USA, 2000.

[8] B. Dougherty, J. White, and D. C. Schmidt. Model-driven auto-scaling of green cloud computing infrastructure. *Future Gener. Comput. Syst.*, 28(2):371–378, Feb. 2012.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[10] Linux. Linux powertop. https://01.org/powertop/. Accessed: 11/01/2013.

[11] Microsoft. Joulemeter. http://research.microsoft.com/. Accessed: 07/01/2013.

[12] I. M. Murwantara and B. Bordbar. A simplified method of measurement of energy consumption in cloud and virtualized environment. *Accepted at Sustaincom 2014 Conference*, 2014.

[13] Oracle. *The Java EE 6 Tutorial*. Oracle Corp, 2013.

[14] M. Shepperd and S. McDonell. Evaluating prediction systems in software project estimation. *IST*, 54(8):820–827, 2012.

[15] N. Siegmund. *Measuring and Predicting Non-Functional Properties of Customizable Programs*. Phd theses, University of Magdeburg, Germany, November 2012.

[16] K. Tatroe, P. MacIntyre, and R. Lerdorf. *Programming PHP*. O'Reilly Media, 2013.

[17] W. Vereecken, D. Colle, B. Vermeulen, M. Pickavet, B. Dhoedt, and P. Demeester. Estimating and mitigating the energy footprint of icts. Report of meeting of Focus Group on ICT and Climate Change on International Telecomunication Union, 2008.