2014 IEEE Fourth International Conference on Big Data and Cloud Computing

A Simplified Method of Measurement of Energy Consumption in Cloud and Virtualized Environment

I Made Murwantara, Behzad Bordbar
School of Computer Science
University of Birmingham
Birmingham B15 2TT, UK
Email: {ixm249, b.bordbar}@cs.bham.ac.uk

***Abstract*—Measuring energy consumption is an essential step in the development of policies for the management of energy in every IT system. There is a wide range of methods using both hardware and software for measuring energy consumed by the system accurately. However, most of these methods measure energy consumed by a machine or a cluster of machines. In environments such as Cloud that an application can be built from components with comparable characteristics, measuring energy consumed by a single component can be extremely beneficial. For example, if we can measure energy consumed by different HTTP servers, then we can establish which one consumes less energy performing a given task. As a result, the Cloud provider can provide incentives, so that, application developers use the HTTP server that consume less energy. Indeed, considering size of the Cloud, even a small amount of saving per Virtual Machine can add up to a substantial saving.**

In this paper, we propose a technique to measure energy consumed by an application via measuring energy consumed by the individual processes of the application. We shall deal with applications that run in a virtualized environment such as Cloud. We present two implementations of our idea to demonstrate the feasibility of the approach. Firstly, a method of measurement with the help of Kernel-Based Virtual Machine running on a typical laptop is presented. Secondly, in a commercial Cloud such as Elastichost, we describe a method of measuring energy consumed by processes such as HTTP servers. This will allow commercial providers to identify which product consumes less energy on their platform.

***Keywords*-cloud, virtual machine, energy.**

I. Introduction

Measuring energy consumption is important because it relates to the cost to run a system. For example, data centres that run Cloud, host thousands of servers and consume large amount of energy. In year 2016, the global demand for energy used by the data centre industry is predicted more than 50 GigaWatt [1]. Considering the ongoing increase in the cost of energy and to take action on the global warming, efficient management of data centres is an active area of research [2], [3], [4], [5]. Measurement is an essential part of planning and management of energy consumption.

Some techniques for the measurement of energy [2], [3], [4], [5] make use of hardware devices commonly known as Wattmeters. Example of such hardware devices is WattsUp [6]. It is also possible to use software products such as Powertop [7] and Joulemeter [8] to measure energy usage of a running system. Most of these measurement methods use workload to simulate the users' activities when they interact with the system under test. However, there is a problem in the current measurement methods. These methods are not designed for measuring energy consumption of individual processes of a running system. In a virtualized environment, we cannot use Wattmeter devices to measure energy consumption per process of a Virtual Machine (VM) because the resources are shared and virtual machines can move from one host to another. As a result, we need a method of measurement that is not reliant on Wattmeter devices.

Our measurement technique is software-based and combines software tools for measuring energy [7], [8] and the workload tools [9]. For example, we can measure energy consumed by different HTTP servers to find out which one is more efficient on energy usage and make a recommendation based on the measurement results.

To show the feasibility of our techniques, we have implemented them on two different platforms: a typical laptop, and commercial Cloud in Elastichost [10]. The suggested method is examplified with the help of the measurement carried out on HTTP servers. The measurement outcome is consistent between Laptop and the Commercial Cloud used in this project.

The remainder of this paper is as follows. In the next section, we present essential background material on the existing power measurement techniques, workload models, software measurement of energy and workload tools. Subsequently, in section three, we describe the contribution of our research. In section four, the measurement results in the Laptop and the Commercial Cloud environment will be explained. Then, section five will discuss the related work. Finally, this paper concludes in section six.

II. Background

In this section, we shall review existing energy measurement techniques.

978-1-4799-6719-3/14 $31.00 © 2014 IEEE
DOI 10.1109/BDCloud.2014.47

A. *Measuring Energy Consumption*

The measurement of energy consumption in the Information Technology is an active area of research [5], [11], [12], [13]. Kansal [12] proposed a tool [8] to monitor the resource usage of runtime software components. There is a wide range of hardware devices to measure energy consumed by a computer such as WattsUp [6]. These devices are commonly known as Wattmeters. Chen et. al. [13] measures virtual machines' energy consumption using Wattmeters. In their measurement, they simulate the users via a workload tool.

Most existing research involving using Wattmeters measure total energy usage of the computer system. Measuring energy consumption in Cloud is non-trivial due to the shared infrastructures and migration of virtual machines [14]. In a shared infrastructure, it is difficult to isolate each virtual machine and measure the energy consumed by it. In addition, virtual machines can be allocated to other hosts dynamically to allow load-balancing within the Cloud. As a result, in the virtualized environment such as Cloud new methods of measurement of energy consumption must be found.

B. *Measurement of Energy Consumption via Software*

There are number of software applications for measuring power consumption in a computer system [15], [7], [8], [16]. Some of these software products use the battery drainage to measure how much energy is consumed. Among of these tools are Joulemeter [8] and Powertop [7]. Joulemeter [8] developed by Microsoft Research measures the specific running services energy consumption in real time. It estimates power usage of computer system by tracking down CPU utilization, Monitor brightness and Disk usage. Joulemeter can measure energy usage of typical laptop on battery mode and other infrastructure using Wattmeter device. Powertop [7] is the default power meter that is deployed with Linux operating system and is built as an Open Source Software. Powertop estimates the power consumption with options using Wattmeter device such as WattsUp [6] or battery information such as the battery lifetime that gives the amount of energy that is consumed by the running processes.

Another group of tools do not rely of measuring the drainage of battery. These tools include Intel Energy Checker [15] and VMMark [16]. The Intel Energy Checker measures energy usage of a computer system on Intel Processor architecture and access the processor information using an API (Application Programming Interface). The VMMark [17] measures power consumption by utilizing the Power Temperature daemon [16], which does the collection of measurement data from the Wattmeter and temperature devices that installed in the cluster machine.

C. *Workload Tool*

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 [9] to generate workload into a system and using the power meter software to measure how much energy is consumed under the given load.

For the rest of this subsection, we explain how Jmeter produces workload. The workload scenario is implemented as script in the XML format and generates load to simulate user's activity. Subsequently, the success and failure of this activity will be reported back to the workload tool to generate the report.

The workload tool implements the traffic model as a way to simulate a client application such as Web browser accessing an HTTP server to execute tasks involved in each scenario. In this paper, the Traffic Model activity mimics a user navigating the request to a HTTP server such as when the user arrives and how requests are generated. According to Liu [18] "a sequence of clicks of a user on the hyperlinks of the same server" can be described as session where the start time of a session will relate to the arrival time.

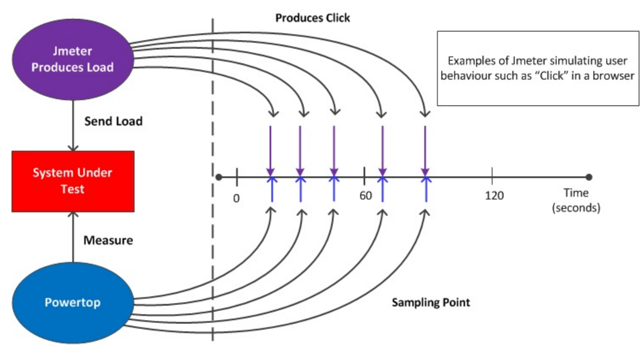

Figure 1. Workload and Measurement

As illustrated in Figure 1, Jmeter [9] generates load to the system by simulating the end user Click of Web browser such as Mozilla. At the same time, Powertop [7] measures the power consumption of all processes. Each Click that produced by the workload tool will be responded by the HTTP server where numbers of Click represent the Client accessing a web page. As a result, when the number of Clients increases, the performance of the HTTP server to address the Clients request will affect the power usage. Upon doing so, the energy consumption will increase as the processes that relate to the HTTP server will take longer time and higher utilization level.

III. Contribution

The contribution of this paper is in three parts. First, we propose a measurement procedure of energy consumption in the virtualized environment for individual processes. So that energy usage of an individual process of an application can be captured. As an example, Nginx [19] an HTTP Server - that has processes such as Master and Worker. Under a given workload, each one of these processes consumes different amount of energy. Our method allows measuring energy that is consumed for those individual processes that have load from a workload tool [9]. As a result, we will be able to select features in the software application that consumed less energy, to be included into a virtualized environment. The second contribution is the measurement technique in the Kernel-based virtual machine (KVM) [20] within the laptop environment, where the software application for measuring energy monitors the electric current drainage from the battery. Calculating energy consumption for laptop is important in saving energy. Imagine finding out which web browser consumes less energy for a given task, such as checking email, will help users to decide on a suitable choice of browser for example during travel, when it is not convenient to recharge batteries. The third contribution is the measurement technique of energy usage in the Cloud environment where the implementation of Advanced Control and Power Interface (ACPI) [21] enables us to monitor the power usage in a Cloud system. As a result, Cloud providers can find out the product that consumes less energy from a number of given software products with comparable functionalities

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

A. *Measurement Procedure*

We aim to measure energy consumption for individual process of an application in a Virtualized environment. We categorize the activity of measurement into power measurement and workload generation. In the power measurement phase, the software application for measuring energy is set up to measure power usage of all processes within the system under test. For the workload generation, the type of load is set up, based on the application under test, such as Database Management system and HTTP server.

As shown in Figure 2, the workload tool sends a given workload to the application inside the virtual machine. The load simulates, for example, users accessing a web page within an HTTP server. At the same time, the software application for measuring energy records the energy consumption in the system under test.

We implement scenario in the workload tool to perform several activities in a consecutive way. As shown in Figure 2, (A) we start the software application for measuring energy. After that, (B) the Workload tools (act as a Client) send a request to the system under test, for example to access web pages and files in the Web server. At the same time, the software application to measure energy captures the power usage of each process that is running in the virtual machine. Subsequently, (C) the workload tool and the software application for measurement generate their reports. The software application for measuring energy reports the amount of power that is consumed per individual process that is generated every one second. The measurement ends (D) when all the reports are generated.

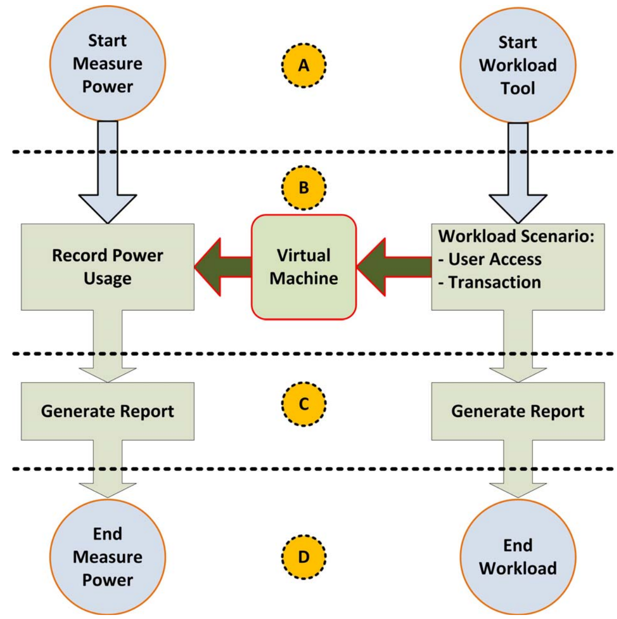

Figure 2. Power Measurement Procedure

In this section, we discussed the procedure for measuring energy consumption in a virtualized environment using the software application for energy measurement and workload tool.

B. *Kernel-Based virtual machine on Laptop*

Consider a workload as described above. In our experiments, we use Ubuntu Server x64 and Windows 7 environments to find out the result of power measurement on different operating systems. In both cases KVM is installed on Laptop.

We take advantage of the ACPI (Advanced Control and Power Interface) [21] to have the real time power consumption from the battery. To standardize the software application for measurement, we calibrate it against the current drainage of laptop's battery for several minutes. The outcome of the calibration is saved into a file called Power Reference.

In the KVM-based laptop, KVM is installed on top of the Ubuntu Server. The system under test gets their workload from a Ubuntu Server, and measures the power usage using Powertop [7] for Linux and Joulemeter [8] for Windows 7 environments. The software application for measurement is running within each system under test.

C. *Measurement on Cloud*

In Cloud, the software that measures the power usage relies on the Advanced Control and Power Interface (ACPI) [21], a platform-independent interface, which is usually used by the Cloud system to turn-off and turn-on the virtual machines [22]. As explained in previous section, APCIs exist both in Cloud and Laptop but are implemented and used differently. When the software application to measure energy runs at the first time, it reads the Power Reference File, and then captures the virtual environment information such as the number of virtual processors and memory capacity. After that, the software application to measure energy will renew its power reference file and become ready to measure the energy consumption per process. The Elastichost [10] Commercial Cloud computing environment that we use as the system under test uses KVM for their virtual machine, similar to the one on the laptop. Snapshot of measurement is shown in Figure 3, the indication of virtualized environment can be found in the virtio0-request [23] process. Moreover, the Kernel updates will not affect the measurement as long as the software application for measurement [7] is not updated.

The measurement in commercial Cloud computing uses similar scenario that is implemented in the Laptop environment. We use Powertop [7] as the software application for measuring energy. All the measurement and the workload are running in Ubuntu Server 12.10 x64 which are installed on Elastichost.

IV. Measurement Results

Now, we explain the application of our method to the measurement of energy for HTTP Server within the virtualized environment and Cloud system. Our method can be easily adapted to measure other platform such as Mobile System and TabletPC. All virtual environments in these measurements use similar configuration that is set up with one Processor and one Gigabyte of memory. In the Elastichost Cloud the processor is the AMD Opteron Processor 6128 2GHz and in the Laptop is Intel Core i5 M520 2.4GHz. During the power measurement preparation, we found that the cache option in the HTTP server increase the power consumption initially because it always create cache file when load hit the web page. In all our experiments, cache is disabled to eliminate the effect of choosing cache algorithm and cache related parameters.

Our method can show the amount of energy consumed by each individual process within a virtual environment such as a Laptop or on Cloud. For example Figure 3 depicts a sample of energy usage by different processes for example can be seen *Virtio0-request* is active for 221.1 μs/Second is consuming 9.47miliWatt. We use the software for sampling over a few period of time and to calculate average energy consumed. We also change the load to measure the amount of energy under different load. We shall examplify that in the next section.

A. Load Model

To produce a load for the system under test, there is a wide range of choices, as explained in section II. Here we make use of Jmeter [9]. We have conducted the measurement by producing the same load for all environments. We have divided the experiment into blocks of four minutes. In the first four minutes we product load of one user per second, and the second four minutes period we produce the load of two users per second and so on. This results in 240, 480, 720, 960 and 1200 users per period of 4 minutes.

```
 102 mW  ,   2.4 ms/s, , , , , Process, inet_gethost 4 ,
49.1 mW  ,   1.1 ms/s, , , , , Process, [migration/0],
37.2 mW  ,   0.9 ms/s, , , , , Process, [kworker/0:2],
31.8 mW  ,   0.7 ms/s, , , , , Interrupt, [9] RCU(softirq),
31.0 mW  ,   0.7 ms/s, , , , , kWork, e1000_watchdog,
21.0 mW  , 489.9 us/s, , , , , Interrupt, [1] timer(softirq),
15.0 mW  , 349.8 us/s, , , , , Process, [ksoftirqd/0],
9.47 mW  , 221.1 us/s, , , , , Interrupt, [41] virtio0-requests,
5.10 mW  , 119.1 us/s, , , , , Interrupt, [11] eth0,
```

Figure 3. Power per Process in Linux Virtualized Environment

For example Figure 4 plot the energy consumed by the Nginx HTTP server in Elastichost Cloud. The thick line is the average within every four minutes block. Figure 5(a) shows the same data as a boxplot graph. These graphs are produced using Statictic software R [24].

One application of our method is when a software product can run on an in-house system or Cloud. We can compare the energy consumption between when application run on an in-home system such as Laptop and Cloud, as explained in the next section.

B. Comparing Power Consumption in Different Virtualized Environment

Measurement of power consumption in the Elastichost Cloud computing system and Virtualized environment is comparable because both systems using similar virtualized technology that is Kernel-based virtual machine (KVM). And, the power consumption is measured based on the Advance Control and Power Interface (ACPI) that exist in Laptop to manage the battery and in the Cloud system to turn-on or turn-off the virtual machines.

As depicted in Figure 5, when smaller load is used, laptops consumes less energy than Cloud. But when the load increases, for example at 3 users per second (720 users accessing in a 4 minutes interval) Cloud consumes less. This trend continues when the load increases. For example, at 5 users per second (1200 users accessing the servers in a 4 minutes interval) the laptop consumes approximately 50% more energy.

In addition, we observe that the pattern of energy consumption in Cloud as the number of users increase becomes more stable, i.e. in Cloud we observe fewer spikes. This can be seen in the outliers depicted on Boxplot graph in Figure 5 .

C. Comparison of HTTP Server Energy Consumption in Cloud

The configuration of application in the Cloud system should also include the cost of operation as an important aspect. For example, the Cloud system designer must know

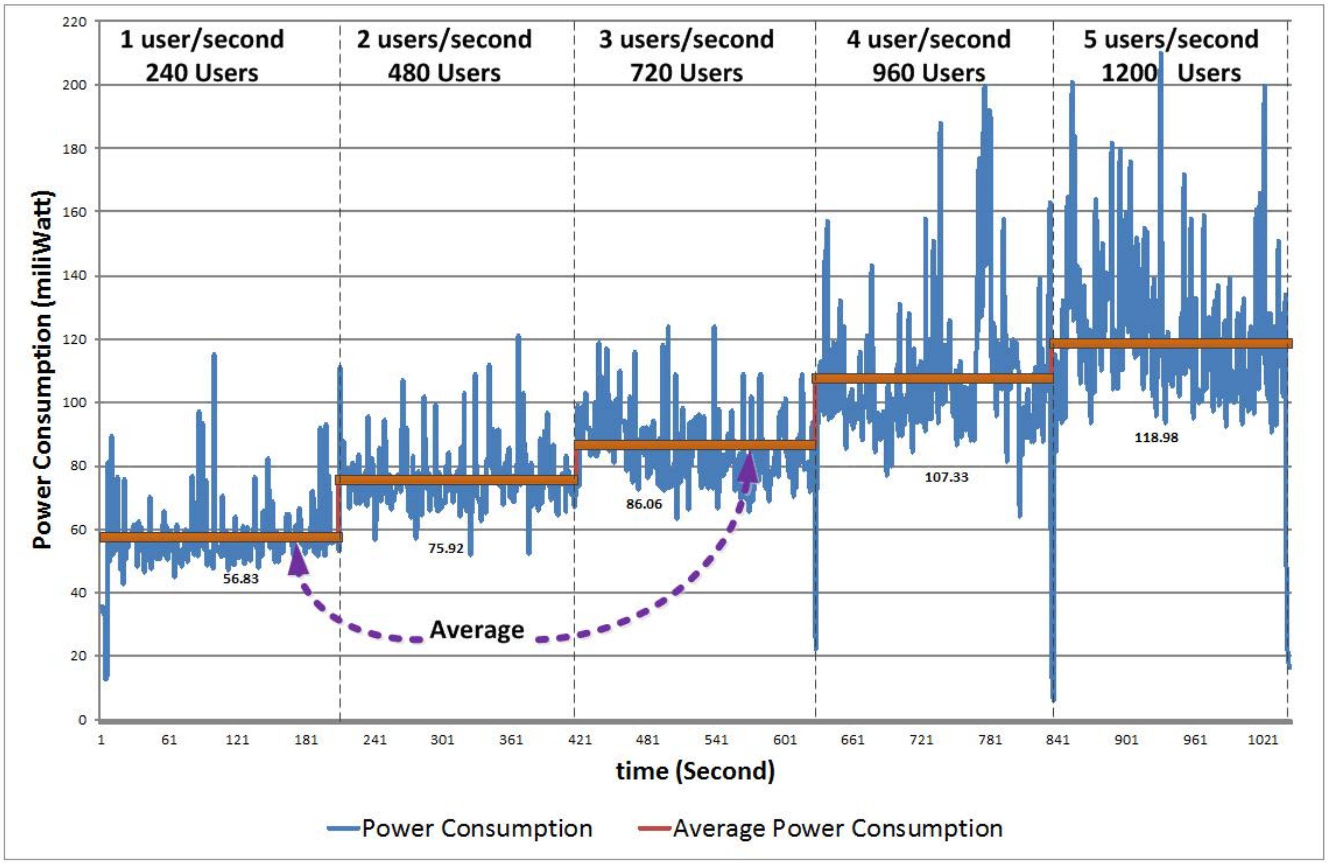

Figure 4. Elastichost Cloud Environment - Nginx Power Consumption

the cost of running HTTP servers from the power consumption point of view.

We measure the power consumption of HTTP server that are listed in the top ten HTTP server [25], which includes Apache, Nginx and Lighttp in it, and measured them within a Commercial Cloud system. As shown in Figure 6 and Table I, the measurement results show us that when 1 user per second arrive in the HTTP server, Apache and Lighttp are more efficient that in average consume less than 50milliWatt. However, when 5 users per second arrive in the system, Apache and Lighttp in average consumed more than 150miliWatt. Meanwhile Nginx power consumption for the arrival of 5 users per second consumed 118.98miliWatt. Lighttp is the lowest in consuming power when less than 3 users arrive to the system per second that is less than 71miliWatt. However, this amount increases higher than Nginx when three or more users arrived to the system. As shown in the Figure 6 (a) and (c), Nginx and Lighttp for 3 users load have similar maximum outliers, however, Nginx has less minimum outliers compares to Lighttp. It shows us that Lighttp may not have constant service when 3 users arrive to system, as a consequence when more users arrive, we speculate, it will accumulate the unfinished service and increase the power consumption.

In the measurement within Cloud system, the increment average of power consumption such as the increment of power usage from one user per second to two users per second load, gives us the information of how the HTTP server will consume energy when more users arrive to the system. From the measurement results, the highest average of power usage increment is Apache that consumed in average 31,63miliWatt, followed by Lighttp that consumed 29,57miliWatt. Nginx has the lowest power usage increment per user per second that is 50% lower than the Apache power increment expenditure.

To confirm the reliability of our measurement, further experiments on different Cloud platform are required.

V. Related Work

Tsung [26] is a workload tool that simulates the load by sending simultaneous users to the system under test. It creates workload in a scenario that is implemented as script in the XML format and generates load to simulate user activity, based on Poisson distribution [27]. Tsung implements the traffic model as a way to simulate a client application such as Web browser accessing an HTTP server to execute tasks

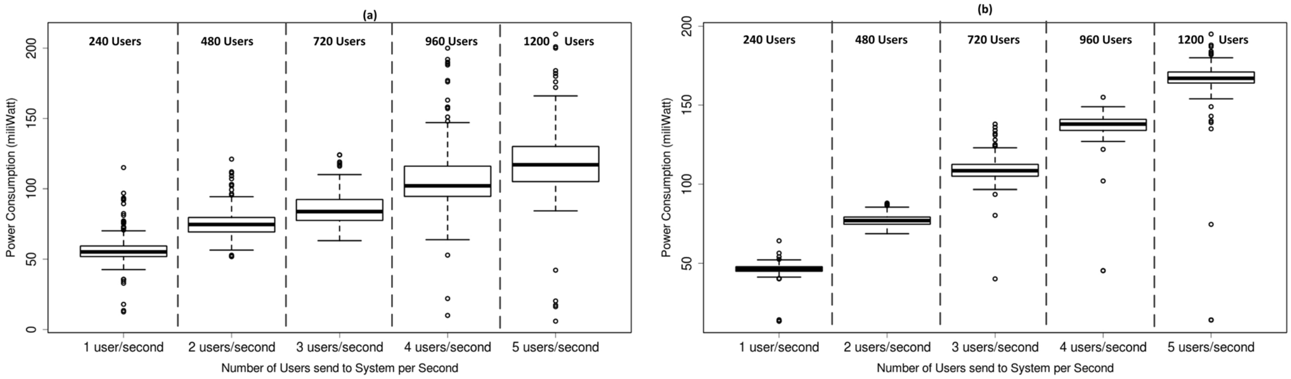

Figure 5. Nginx Web Server Power Consumption in two different environments - (a) Elastichost Cloud System and (b) Ubuntu KVM in Laptop

Table I
AVERAGE POWER CONSUMPTION IN UBUNTU AND WINDOWS 7 IN THE KVM OF A LAPTOP AND ELASTICHOST CLOUD USING DIFFERENT LOAD

OS & App.	1 User*	2 User*	3 User*	4 User*	5 User*
Ubuntu Server					
Apache	44	84.49	122.79	161.15	198.54
Nginx	46.09	77.25	109.22	136.70	165.14
Lighttp	37.03	73.76	109.27	145.33	181.42
Windows 7					
Apache	994.16	1273.75	1597.5	1903.33	2115
Nginx	3775	617.08	852.91	1073.75	1291.66
Lighttp	374.16	655.41	890.41	1077.91	1359.61
Elastichost Cloud					
Apache	48.19	84.77	109.40	136.83	174.72
Nginx	56.83	75.92	86.06	107.33	118.98
Lighttp	36.48	70.27	96.98	127.08	154.77

** All measurement results in miliWatt*

involves in each scenario. In our measurement techniques, we use Jmeter [9] to send workload to the system under test where the scenario of load can be setup simpler and more controlled fashion than Tsung. Jmeter has capability to constantly send load to the HTTP server such as one user per second for four minutes without interruption, so that, we can measure the energy consumption based on the number of users that arrived in the system as the workload. For example when 5 users per second arrive, the system is accessed by 1200 users during the four minutes.

Kanzal [12] et. al., developed a technique to measure power consumption of a virtual machine using Joulemeter [8]. Kanzal [12] developed a power model to present the VM power consumption. The power model takes elements such as CPU, Memory, Disk and other hardware components as the main measurement unit. A software application, most of them, have more than one process running at the same time. For example an HTTP server may run the cache and logging status concurrently. However, Kanzal [12] method only measures one process at a time, so other processes that associated with a software under measurement cannot be included. As a result, total energy caused by using an application is not measured. In this paper, we mostly deal with Linux based systems.

We have also extended our measurement to HTTP servers [28], [19] that are running within Windows 7 on a VM environment of a Laptop. In this measurement, the load generated in this system is similar to Ubuntu in KVM Laptop. As shown in Figure 7, the power consumption of Apache HTTP server when one user per second arrive to the system in average is 994.16miliWatt and increase to 1273.75miliWatt when two users per second arrive to system. We take this measurement to prove that our method also applicable to Windows environment. It can be seen that the amount of energy consumed under Windows 7 is much more higher than Linux, when we use the same load, compare Figure 7 (unit is Watt) and Figure 6 (unit is miliWatt).

VMWare VMMark [29] measures the performance of the Virtualized system as a collection of virtual machines executing a set of diverse workloads. In VMMark, the large portion of utilization on the CPU, disk, network and memory subsystem influence the measurement of energy. VMMark measures total energy consumption per cluster machine using Wattmeter and Thermal devices that are installed in the machines, which are collected using a software application data collector. In our measurement technique, we use diverse workloads on different HTTP servers. We also looked at browsers and different Database Management Systems; due to space restriction those results not included. Further, energy consumption of individual processes of a software application that is being measured in the virtualized environment can be captured. So that, we can analyse the best configuration of a virtual machine that consumes less energy.

In our experiments, the load limitation is the maximum number of open files allowed for a user in the operating system. For example Ubuntu allowes 1024 files in default configuration. In this experiment, the workload tool gener-

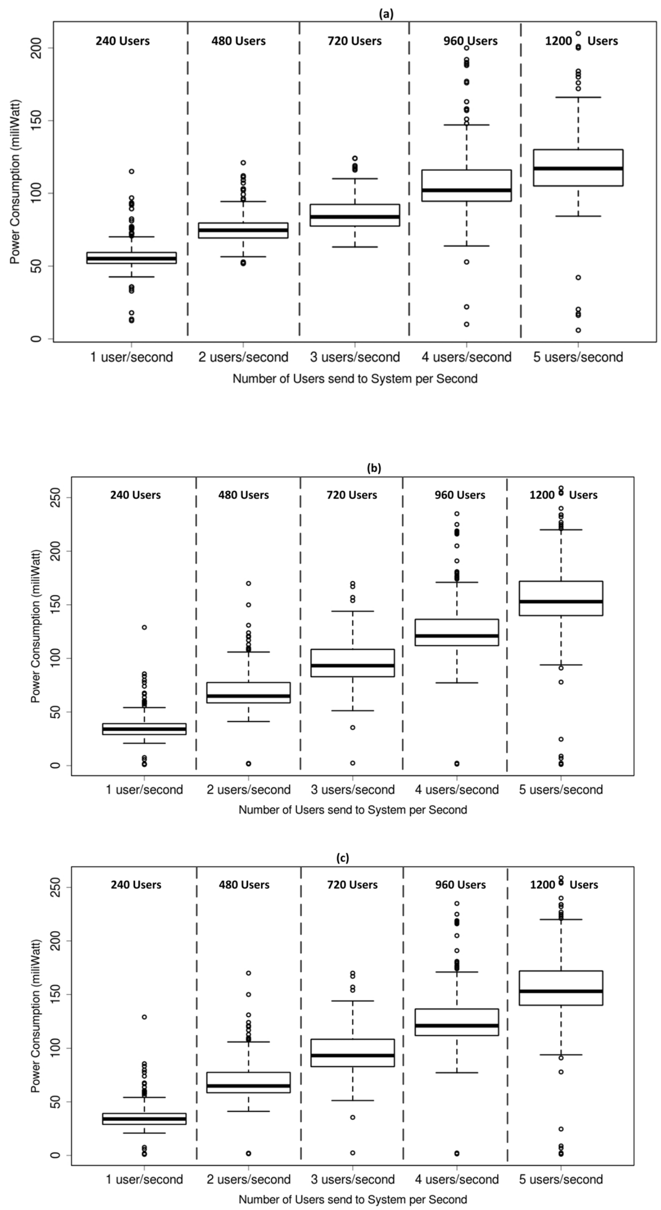

Figure 6. Elastichost Cloud - Nginx (a), Apache (b) and Lighttp (c) HTTP Server Power Consumption

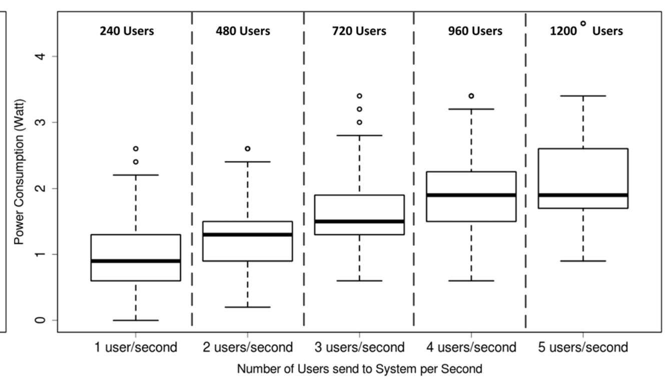

Figure 7. Windows 7 in KVM - Apache Web Server Power Consumption

ates maximum of 300 users per minute that is below the operating system default limit configuration.

Increasing the load on the system leads of higher energy consumption. In our experiments cache options were disabled. Turning the cache on and choice of suitable caching algorithms are options for future research.

VI. Conclusion

This paper proposes a method of measurement of energy consumption in virtualized environment such as Cloud or laptops with virtualization support. The suggested measurement method involves using of workload tools to simulate users accessing the system and software products such as Jmeter and Powertop for calculating amount of energy used. We show that it is possible to measure energy consumed by indvidual processes, components, servers and applications. When more than one application can perform a give functionality, this technique will allow the users to choose the application which requires least amount of energy. Similarly, by identifying such an application, the cloud providers can encourage the users and developers to use applications and software components that consume less energy. To show the viability of our methods, we have measured energy consumed by a number of HTTP servers on a typical laptop and elastic host. Under the given load, we infer that Nginx consumes less energy compared to other competing HTTP servers such as Apache and Lighthttp servers. Also as expected, our method confirms that running applications on Cloud is more economic than running on laptops. Further experiments on different laptops and Cloud platforms are required to confirm our result. However our techniques can be easily adopted for such experiments.

ACKNOWLEDGMENTS

First author was supported by the Secretary General of Higher Education (DIKTI), The Ministry of Education and Culture, Republic of Indonesia, Project No: BLN120200-DIKTI 2012. Second author wishes to acknowledge generous financial support from HP Lab.

References

[1] 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.

[2] P. Xiao, Z. Hu, D. Liu, G. Yan, and X. Qu, "Virtual machine power measuring technique with bounded error in cloud environments," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 818 – 828, 2013.

[3] Z. Zhang and S. Fu, "Profiling and analysis of power consumption for virtualized systems and applications," in *IPCCC IEEE, 2010*, 2010, pp. 329–330.

[4] Y. Li, Y. Wang, B. Yin, and L. Guan, "An online power metering model for cloud environment," in *Network Computing and Applications (NCA), 2012 11th IEEE International Symposium on*, 2012, pp. 175–180.

[5] A. Bohra and V. Chaudhary, "Vmeter: Power modelling for virtualized clouds," in *IPDPSW 2010*, april 2010, pp. 1 –8.

[6] Wattsupp, "Watts Up Power Meter," http://www.wattsupmeters.com, accessed: 20/01/2013.

[7] Linux, "Linux powertop," https://01.org/powertop/, accessed: 11/01/2013.

[8] Microsoft, "Joulemeter," http://research.microsoft.com/, accessed: 07/01/2013.

[9] Apache, "Apache jmeter," http://jmeter.apache.org, accessed: 04/01/2014.

[10] Elastichost, "Iaas Architecture," http://www.elastichosts.co.uk/, accessed: 16/04/2013.

[11] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, and Y. Chen, "Greencloud: a new architecture for green data center," in *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, ser. ICAC-INDST '09. New York, NY, USA: ACM, 2009, pp. 29–38.

[12] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *Proceedings of the 1st ACM symposium on Cloud computing*, ser. SoCC '10. New York, NY, USA: ACM, 2010, pp. 39–50.

[13] Q. Chen, P. Grosso, K. van der Veldt, C. de Laat, R. Hofman, and H. Bal, "Profiling energy consumption of vms for green cloud computing," in *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, dec. 2011, pp. 768 –775.

[14] R. Buyya, A. Beloglazov, and J. H. Abawajy, "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," *CoRR*, vol. abs/1006.0308, 2010.

[15] Intel, "Intel energy checker," http://software.intel.com/en-us/articles/intel-energy-checker-sdk, accessed: 10/01/2013.

[16] V. Makhija, B. Herndon, P. Smith, L. Roderick, E. Zamost, and J. Anderson, "Vmmark: A scalable benchmark for virtualized systems," *Technical Report VMware TR-2006-002*, Jan. 2006.

[17] VMWare, "Vmware product," http://www.vmware.com, accessed: 4/01/2013.

[18] Z. Liu, N. Niclausse, and C. Jalpa-Villanueva, "Traffic model and performance evaluation of web servers," *Perform. Eval.*, vol. 46, no. 2-3, pp. 77–100, Oct. 2001.

[19] Nginx, "Nginx," http://nginx.org/, accessed: 5/01/2013.

[20] IBM, "KVM Open Virtualization," http://www-03.ibm.com/systems/virtualization/, accessed: 19/04/2013.

[21] J. Kolinski, B. Press, R. Chary, and A. Henroid, *Building the Power-Efficient PC: A Developer's Guide to ACPI Power Management with CD-ROM*. Intel Press, 2001.

[22] Elastichost, "Shutdown, hard power off and hard reset button - http://www.elastichosts.com/question/ what-do-the-shutdown-hard-power-off-and-hard-reset-buttons-do/," accessed: 01/10/2013.

[23] KVM, "Virtio," http://www.linux-kvm.org/page/Virtio, accessed: 25/09/2013.

[24] R, "The r project for statistical computing," http://www.r-project.org, accessed: 4/01/2014.

[25] T. Corp., "Web servers market position report," http://w3techs.com/technologies/market/webserver/10, accessed: 25/11/2013.

[26] Tsung, "Tsung user's manual," http://tsung.erlang-projects.org, accessed: 10/01/2013.

[27] a, "Statistical analysis handbook," http://www.statsref.com/HTML/index.html?poisson.html, accessed: 10/03/2013.

[28] Apache, "Apache - http server project," http://httpd.apache.org/, accessed: 5/01/2013.

[29] VMWare, "Vmmark: Virtualization benchmark," http://www.vmware.com/products/vmmark, accessed: 25/09/2013.