Energy Consumption Estimation of Virtual Machines

Ingolf Waßmann, Daniel Versick, Djamshid Tavangarian
Research Group Computer Architecture
University of Rostock
Joachim-Jungius-Str. 9
18059 Rostock, Germany
[firstname.surname]@uni-rostock.de

ABSTRACT

Energy consumption of IT increased continously during the last decades. Numerous works have been accomplished for improving energy efficiency of hardware whereas software energy efficiency has been ignoried for a long time. This contribution presents a novel approach for estimating energy consumption of applications in different execution environments. The system is the basis for automatic optimization of software execution in an energy-efficient way by finding the best-suiting host computer. Thus, it opens novel ways to further improve energy-efficiency of IT systems. Exemplary, the approach is tested in a virtualized data center environment, where virtual machines are the applications. The presented approach is a vehicle for automatically finding the most energy-efficient host machine for any virtual machine system.

The paper presents a new algorithm for estimating virtual machine power consumption and shows the accuracy of the presented approach by means of measurements.

Categories and Subject Descriptors

D.4.8 [**Operating Systems**]: Performance—*Modeling and prediction*; K.6.2 [**Installation Management**]: Benchmarks, Computer Selection, Computer Equipment Management, Performance and Usage measurement

General Terms

Algorithms, Measurement, Performance

Keywords

Energy efficiency, power consumption estimation, virtual machine placement

1. INTRODUCTION

Energy consumption of IT systems constantly increased for decades. The power demand of current data centers is about 1.5 % of worldwide energy production, whereas the growth is tremendous. The energy consumption of data centers almost tripled within the years 2000 and 2010, which corresponds to an average annual growth rate of about 12 % [4, 5]. As energy costs are rising too and governments come under pressure to reduce carbon dioxide production, novel technologies have to be developed to reduce energy consumption of IT systems. During the last years a lot of techniques appeared for improving IT energy efficiency. Most innovations have been arised in the field of mobile computing as mobile systems need to be very energy-efficient for having a longer battery-driven runtime. Currently, many existing techniques focus on improving energy efficiency of hardware, whereas software issues are an important research field as well.

Techniques for improving energy efficiency of software execution can be employed during two phases of software life cycle:

- Code changes towards a better energy efficiency may be used during development and compile time such as reducing the number of memory accesses by e.g. linking parts of the code to scratchpad memory [9].
- Energy consumption of software can be improved during runtime by changing configuration of the runtime environment. This may involve a migration of load to a more energy-efficient execution environment.

For both approaches an energy profiling of the application is mandatory as well as an estimation of energy consumption after change. In the following, current energy profiling and estimation solutions are identified before our novel energy estimation system is presented in an example use case.

In [2] the authors present PowerScope, a combined hard- and software solution for measuring energy consumption of software and software parts, which is used for developing mobile applications. As the paper is an early work, the presented approach is directly only usable for uniprocessor systems. In [8] Snowdon et al. present the Koala power prediction system, which was implemented as part of the Linux kernel. Koala enables the OS kernel to predict the power consumption behaviour when changing core frequency and voltage using Dynamic Voltage and Frequency Scaling (DVFS). For accurate power estimation Koala needs predefined energy consumption models, which are highly architecture-dependent and therefore, have to be assembled for any new processor. In [6] and [7] the authors present measurements for acquiring the power consumption of virtual machines using existing measuring tools. An interesting approach is presented by Kansal et al. in [3]. The authors present a software system for measuring and estimation of virtual machine energy consumption, which is a comparable approach to the presented work in this contribution. Their work includes a power consumption model which, as we show in this contribution, is not generic enough for mapping current processor features.

The present contribution introduces an energy measurement and estimation system for virtual machines with a self-adaptable energy model which acquires accurate results even on current architectures. It also presents implementation facts of the system as part of a virtual machine placement algorithm which finds a virtual machine placement with highly improved energy efficiency. Results of the energy measurement and estimation system, which show its accuracy, are presented in the end of the paper.

2. SYSTEM ARCHITECTURE

The work presented in this contribution is part of a joint project between industry and the University of Rostock. Goal of this project is the improvement of energy efficiency of data centers by enhancing the utilization of physical server systems by dynamic load migration between physical servers within the data center. Depending on a number of load and environment parameters within the data center, the best placement of virtual machines (VMs) and a suitable configuration of the air condition as well as systems of the building automation shall be calculated in a dynamic and holistic way [10]. A promising approach for choosing the most energy-efficient data center configuration from various possible configuration options is the consumption estimation by means of a metric that estimates real energy consumption of server systems as well as data center infrastructure based on current or future load situations.

A first step to solve this complex problem, is power consumption estimation of virtual machines running on different platforms and varying load conditions. For this, it is necessary to acquire the fraction of energy consumption which is consumed by each VM.

Before presenting our approach, it is necessary to introduce the whole system architecture, which is the framework for our developed VM energy consumption estimation mechanism (Figure 1).

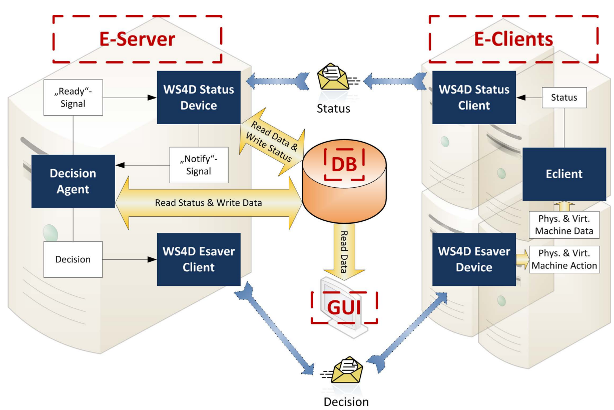


Figure 1: Whole system architecture

Every physical machine of a data center is called *E-Client* in this context. The E-Client includes a Xen hypervisor as virtualization solution and our E-Client software, which periodically acquires the physical machine's system status and sends it by a web service interface [11] to a dedicated server, called *E-Server*. Among other things, the status consists of resource load information assigned to the corresponding VMs, which can fastly be determined by means of the hypervisor. The E-Server stores this data into a database, which holds the overall system load situation. Based on this information the *Decision Agent* calculates an energy-efficient VM placement and triggers VM migrations. The decision-making is actually fully based on hardware loads. In a first attempt VMs from overloaded servers will be migrated to lower loaded machines. If there is no overloaded server anymore, VMs from low-loaded machines are aggregated. Later unused servers may be put into energy-saving modes.

Now this decision algorithm will be extended by energy-based considerations. Thus, it is chosen a target system from a number of possible candidates for a virtual machine migration that has the lowest power consumption when executing this virtual machine. The developed energy consumption estimation mechanism for VMs uses existing hardware and software components to measure the actual VM loads and corresponding whole system power consumption. As the resources processor (CPU), hard disk drive (HDD), and network interface controller (NIC) have an important influence on power consumption, they are observed regarding to the type and duration of its utilization by a certain VM. Furthermore, these resources can be analyzed relatively easy by the hypervisor without needing to install an agent to the guest virtual machines. Other components like memory accesses also influence the energy consumption and will be respected by the presented approach to a certain degree. Based on individual energy models for each VM, their energy consumption as part of the total measured power consumption of the system can be calculated.

3. POWER CONSUMPTION ESTIMATION

Total power consumption of a computer system comprises a CPU (P_C), HDD (P_H), NIC (P_N), and idle part (P_{Idle}) as well as some other components' consumptions which are firstly ignored. The latter corresponds to energy consumption while the system has no load. Thus, the power consumption can be seen as a sum of the partial consumptions of its components. As the consumption of components is not always a simple function of the load, our energy model will approximate the enery consumption by polynomial regression methods with varying degrees. Other approaches for acquiring energy consumption in dependency of load could be table-based but thus, would have a larger memory footprint. Independent variables of our regression function are the observable resource utilizations (CPU: x_C, HDD: x_H, NIC: x_N). This principle of modeling energy consumption is illustrated in Equation 1.

$$\begin{aligned} P_{Sys} &= P_C + P_H + P_N + P_{Idle} \\ &= a_{1,C} * x_C + a_{2,C} * x_C^2 + \ldots + a_{m,C} * x_C^m + b_C \\ &+ a_{1,H} * x_H + a_{2,H} * x_H^2 + \ldots + a_{m,H} * x_H^m + b_H \\ &+ a_{1,N} * x_N + a_{2,N} * x_N^2 + \ldots + a_{m,N} * x_N^m + b_N \\ &+ P_{Idle} \end{aligned} \tag{1}$$

In contrast to our introduced energy model which uses polynomial functions for modelling the energy consumption of any resource, the approach described in [3] assumes a linear correlation, which fails at current computer architectures. Figure 2 depicts measurements, acquired at an AMD Phenom II processor with Turbo Core mode, that clearly illustrates a non-linear dependency between CPU load and energy consumption. Such characterstics occur in many current processors with integrated dynamic overclocking functions, like AMD's Turbo Core or Intel's Turbo Boost. When fully loading only some of the existing cores, the energy consumption increases drastically as the processor overclocks these cores. When having high load at all processor cores, the overclocking feature is deactived as the processor might become too hot.

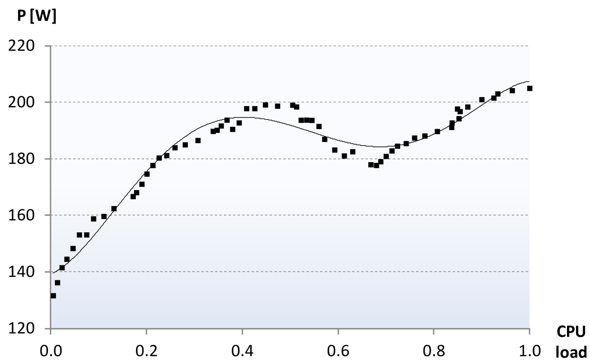


Figure 2: Power consumption depending on CPU load (load increase core by core) on AMD Phenom II with Turbo Core

For estimating the power consumption of virtual machines in dependency of their load, our estimation system acquires system related parameters during two phases. The first phase (*init phase*) has to be started before running any VM on the host (thus, at startup time of the physical machine) and is deployed to evaluate the $a_{i,j}$-parameters in Equation 1 for the physical machine (PM). These *PM specific parameters* specify the power consumption of a physical machine in dependency of its load regardless of any virtual machine running on that physical host. Thus, after the first phase, a rough estimation of power consumption can be calculated as the overall load of a physical host can be estimated to a certain degree by knowledge about the load characteristics of all virtual machines. A more detailed view on power consumption is assembled by constantly measuring the energy consumption of the physical systems in dependency of the load situations within the virtual machines during the second phase (*runtime phase*). Thus, the same parameters, which are acquired for a physical machine during the init phase, are now calculated for any virtual machine on every host (*VM specific parameters*).

In the following, both phases will be looked at in more detail.

3.1 Init Phase

During the initialization phase synthetically varied load is used to determine the systems power consumption in different situations, which is acquired by means of a smart metering system. These values are inserted into polynomial regression according to Equation 1. The degree of the regression (m in Equation 1) is automatically selected. It is chosen the smallest degree which already has an accuracy within a user-defined limit. The result represents the best approximation for the specific system characteristics (see also Figure 2) with a minimized computation effort. The acquired parameters are sent by means of a status message to the *E-Server*, which stores them into a database.

3.2 Runtime Phase

Real VM loads are measured and used to acquire VM specific parameters during runtime phase. This is necessary, because of varying and non-linear workloads in real data center environments which don't match the synthetic load of the init phase. To get specific VM parameters, Equation 1 is modified in a way that it is the sum of individual VM energy values ($i = 1..n$ with n is the number of VMs) (Equation 2).

$$P_{Sys} = \sum_{i=1}^{n}(P_{i,C} + P_{i,H} + P_{i,N}) + P_{Idle} \ . \qquad (2)$$

Whenever a new VM appears on a PM, the resource utilization of each VM, as well as the corresponding total power consumption of the system, will be tracked for a certain time. The tracked data will be employed to calculate the characterizing parameters of each VM by multiple regression methods. The multiple regression computes all unknown regression parameter values in the P_{Sys}-function, which has independent variables for any load type and each virtual machine. As the number of needed measurement values depends on the number of independent variables in a function to be analyzed by multiple regression, it is as high as the number of hosted virtual machines multiplied with the number of observed resources. Furthermore, in case of a high regression error, it is further increased while incrementing the regression degree as already described for the init phase. The polynomial regression includes matrix calculations like multiplication and inversion, which have polynomial (usually cubical) complexity. They can easily be computed by modern processors as the problem sizes are relatively small.

The calculated parameters will be sent to the *E-Server*. After storing in the database, the *Decision Agent* can use these parameters to get a more accurate power consumption estimation while searching for energy-efficient migration decisions. After a certain settling time, the system learned VM-specific parameters for each VM on any PM in the data center. Thus, it is possible to select migration targets with the lowest energy consumption after successful migration.

Experiments show, that learning of VM parameters for many concurrently running VMs works well, which will be shown in Section 4. Furthermore, learned parameters for a certain PM can be used for other PMs with the same or similar properties. This reduces additional overhead.

Because of changes of VM's internal applications or PM's hardware replacements, the parameters must recalculated automatically. This requires, that the estimated total power consumption, based on the sum of certain VM values, will be compared regularly with the measured one. If a predefined error is exceeded, the parameters need to be updated.

The described estimation system was implemented as part of our energy management software in C++ programming language. The following section will show first measurements for getting an impression of the accuracy of the introduced approach.

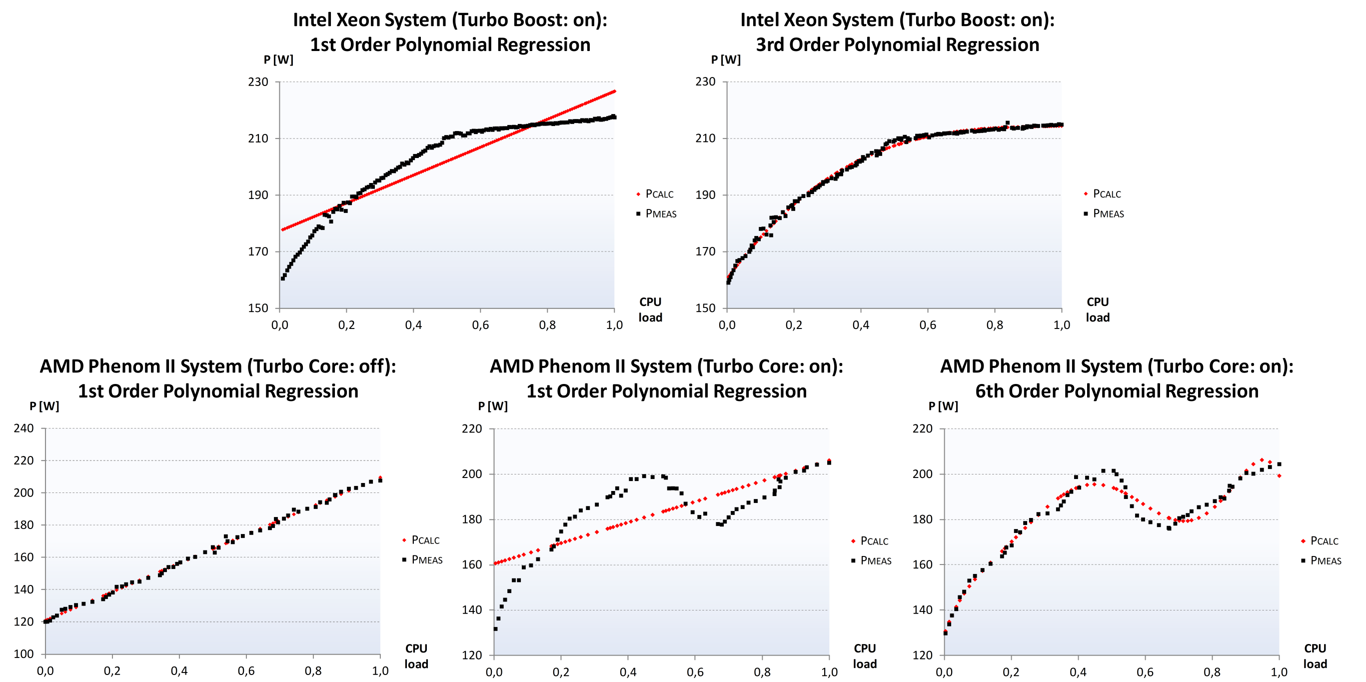


Figure 3: Init phase results

4. RESULTS

Two different test systems were used for the functional test of the developed prototype. The first one is a current Xeon server system with the following specifications: 2 x Intel Xeon CPU E5620 (8 cores @ 2,4 GHz and 16 threads), 16 GB DDR3 RAM, 3 x 1 TB SATA HDD, 2 x 1000Base-T-Ethernet, and Linux operating system Ubuntu Server 11.10 x64.

The second testbed, which will henceforth be referred to as AMD Phenom II system, has the following characteristics: AMD Phenom ll X6 1090T CPU with 6 cores @ 3,2 GHz, 12 GB DDR3 RAM, 1 x 1 TB SATA HDD, 1 x 1000Base-T-Ethernet, and Linux operating system Ubuntu 11.10 x64.

In both systems, the complete project software stack with all necessary packages and libraries, as well as the Xen hypervisor version 4.1 were installed. Any virtual machine had the following configuration: one virtual CPU, 128 MB RAM, 5 GB HDD, and Linux operating system Debian 6.0.3 x64.

For power consumption measurements, the Christ Elektronik network-based power strip *CLM5-IP* was used, which provides the actual total power consumption of the system every second via a network socket.

4.1 Init Phase

Firstly, results of the init phase were analyzed. A workload generator was implemented as part of the init phase software, which increases the CPU load successively. Thus, it first increases the load at one core up to 100 % before loading the next one and so on. For every load situation the power consumption was measured using the network-based power strip. Acquired data was the bases for calculating the above-mentioned correlation parameters as PM system parameters. As 58 % of the dynamic power consumption are produced by CPU load within a computer system [3], we firstly only examined the CPU.

This approach to determine the system parameters was performed on both testbeds. Results of the measurements are shown in Figure 3.

The upper two diagrams visualize the correlation of CPU load and power consumption of the Intel Xeon server system. The non-linear curve can be explained by the many high-performance and energy-optimizing features of the Intel processor. A huge influence has Intel's *Turbo Boost* technology, which dynamically adjusts the clock frequencies according to the current load situation. Furthermore, Intel's *Hyper-Threading* technology ensures an improved multitasking, which affects the energy consumption as well.

For a good approximation of this curve, it's not sufficient to merely use a simple straight line as the upper left curve depicts. Both, the average error, which is 2.6 %, and the maximum error of 10.8 % are too high. The latter in particular can make a difference of up to 20 W of calculated and actually measured energy consumption. A significant improvement of calculating the parameters is realized by a second order polynomial regression. With a maximum error of about 3 % and an average value of 0.6 %, the estimated power consumption can be approximated very well. A further increase of polynomial regression's order improves the error not exceeding 1.9 % and an average value of 0.3 % (regression curve in the upper right diagram in Figure 3). With an average deviation of 0.7 W on the test system, the power consumption can be precisely determined.

The AMD Phenom II processor supports *Turbo Core* technology, which is comparable to Intel's Turbo Boost. While deactivating Turbo Core mode, a simple linear regression is sufficient to calculate system parameters with high precision, because the function can be very well approximated by a straight line (left diagram in bottom line in Figure 3). The maximum error is 2.5 % and the average is 0.7 %.

If Turbo Core mode is active, the function shows another behaviour, as depicted in the other two diagrams of the bottom line in Figure 3. For this purpose, a multiple regression with higher order than three is necessary. If using a fourth order polynomial function, the high maximum error of 22 % (linear regression) and 11.6 % (second order polynomial regression) is improved to 5.5 %. Here, a doubling of the degree halves the maximum error. A sixth order polynomial regression supplies an acceptable precision with maximum error of 3.8 % and an average value of 1.4 %, which is about 1.8 to 2.9 W on test system (bottom right diagram in Figure 3).

Real world applications may show other load characteristics than successively increasing core load as we used (e.g. even load on all processors). To acquire the error of our approach when using load characteristics of any application, additional experiments were done. Figure 4 shows results of measurements which have been acquired when setting the CPU load on the cores to any possible load variation.

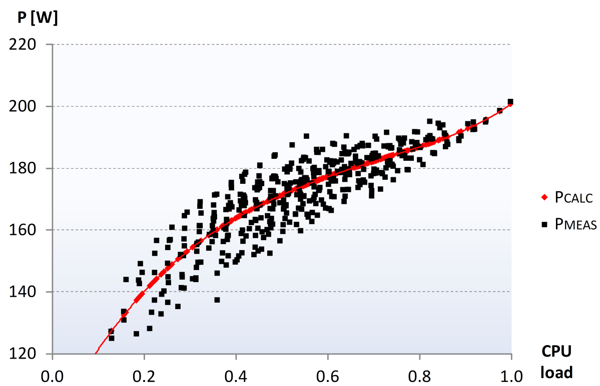


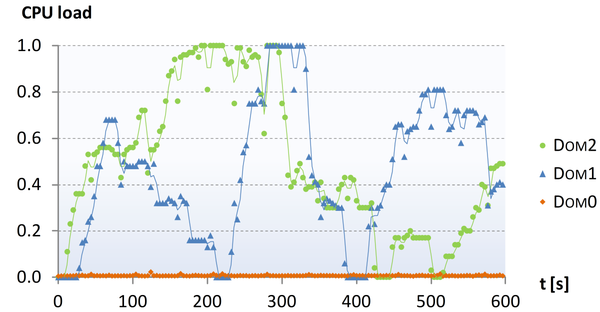
Figure 4: Power consumption in dependenancy of CPU load (all possible core load variations) on AMD Phenom II with Turbo Core

It is shown, that there is a greater spreading of the values, which results in a higher average error of 3.1 % while using sixth order polynomial regression (in contrast to 1.4 %). This is still a very acceptable precision.

4.2 Runtime Phase

The accuracy of our VM tracking mechanism which is used to acquire the fraction of energy consumption of each running VM is evaluated in the second measurements. For CPU load generation the same tool was used, which is already known from init phase. It was executed with randomized data within the VMs (randomized load as well as randomized time spans for load characteristics). In the virtualized environment, both guest VMs (Dom1 and Dom2) and the always running root VM (Dom0) were executed (Figure 5).

Over a period of five minutes the average CPU load of all three VMs and the total power consumption of the system were tracked every five seconds. While the root VM (Dom0) has no significant CPU load during this time (curve is always nearly at 0 % load), both guest VMs were massively loaded. The following polynomial regression then calculated the specific parameters for all VMs simultaneously. P_{MEAS} in the bottom curve of Figure 5 shows the measured system energy consumption, whereas P_{CALC} corresponds to the estimated



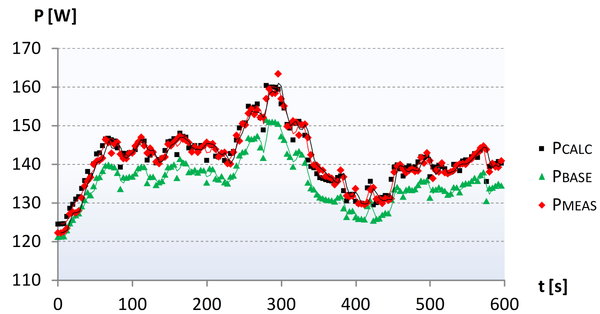


Figure 5: Runtime phase results

values. Both curves are nearly identical. The maximum error is 5.8 % and on average 1.1 %. Every VM with the described properties consumes up to 17 W - 18 W on the test system only because of its CPU load.

To illustrate the necessity of specific learned model parameters for each VM, which is done only during the runtime phase, also the calculated energy consumption on the basis of the initial system parameters is shown in the diagram (P_{BASE}). It can be seen that this estimation always underestimates the actual value at an average of about 4 %, which is about 5 W - 8 W on the testbed.

4.3 Analysis

Each mesurement was repeated 10 times to reduce random error. The shown graphs are based on arithmetic averages. The systematic error is influenced by different sources like measuring faults of the used power strip, the overhead of the E-Client and Xen hypervisor software [1].

Using various experiments the functionality of the developed solution could be demonstrated. Here, three different function types were examined which reflect the relationship between CPU load and power consumption. By using multiple regression, the software automatically could approximate every power consumption function very well.

In order to estimate the energy consumption more precisely, VM-specific parameters are required. Last measurements show the difference between estimation of power with VM specific parameters (P_{CALC}) and without them (P_{BASE}) which differs in about 4 %. This phenomenon can be explained by the loss of performance through virtualization.

In two test cases, it was demonstrated that parameters for multiple VMs can be calculated at the same time. This saves tracking phases in later practical use, and thus reduces online costs. By simultaneous determination of the parameters for three VMs, an average error of only 1.1% was shown on the test systems. Divided among the three VMs, the energy consumption of each domain can be estimated with an error of less than 0.4 %, which only corresponds to 0.5 to 0.8 W on the test machine. This result is very promising in terms of the described application of energy-efficient VM migrations.

5. CONCLUSION AND FURTHER WORK

This contribution presented a novel approach for estimating energy efficiency of virtual machines on different hardware platforms and thus, can be used to automatically find the best-suiting execution environment for VMs. It has been shown that the results are very accurate as the average error is only about 1.1 %. For further enhancements, additional tests with other hard- and software configurations may be performed. Furthermore, the tests should be executed with real world workloads, wherein the energy savings by energy-efficient migrations can be determined.

The developed concept provides a solution for the power capping problem in virtualized data centers. Power capping is a common technique in data centers to improve energy-efficiency. By hardware-based throttling of the performance of individual servers, the total power requirement is reduced. Thus, there is further additional energy for other systems. Due to the properties of server virtualization, this method is not directly applicable in virtualized environments, because reducing performance of a whole PM includes reducing performance of each hosted VM. It would be better to cap every VM's power consumption instead of the whole server. The presented approach can estimate VM power consumption, which is the basis for solving this problem.

Furthermore, the developed solution allows the measurement of influence of certain hardware components to total energy consumption. Based on this information, the resource utilization may be optimized. If a specific device consumes much power, its use may be restricted. From this point of view, a new payment system for cloud environments, based on the resulting measured energy consumption by user's VM, may be established.

It's not only possible to determine the energy behavior of hardware resources, but also of individual VMs and therefore of individual applications. This allows to create customized power profiles for VMs, groups of similar applications, or even users. The energy profiles of applications can provide conclusions for the software developers to optimize the energy performance of their programs.

As shown, the developed solution provides a variety of approaches to significantly improve energy-efficiency of IT systems. In addition to the optimization of each individual IT system, the presented approach creates a novel way for an additional energetic and associated financial savings potential.

6. ACKNOWLEDGEMENT

This work is part of the project "Improving Energy Efficiency of Data Centers" which was funded by the Ministry of Economics, Labor, and Tourism of the German Federal State Mecklenburg Western-Pomerania under grant number V-630-00008-2010-/044/047 by financial means of ESF and ERDF. We thank the funding authorities and our industry partner INR - Informationssysteme Rostock for their kind support.

7. REFERENCES

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, SOSP '03, pages 164–177, New York, NY, USA, 2003. ACM.

[2] J. Flinn and M. Satyanarayanan. Powerscope: a tool for profiling the energy usage of mobile applications. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 2–10. IEEE, 1999.

[3] 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*, SoCC '10, pages 39–50, New York, NY, USA, 2010. ACM.

[4] J. G. Koomey. Worldwide electricity used in data centers. *Environmental Research Letters*, 3(3):034008, July 2008.

[5] J. G. Koomey and D. Ph. GROWTH IN DATA CENTER ELECTRICITY USE 2005 TO 2010. *The New York Times*, 2011.

[6] M. Marcu and D. Tudor. Power consumption measurements of virtual machines. In *Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on*, pages 445–449. IEEE, 2011.

[7] M. Marcu, D. Tudor, and S. Fuicu. Power consumption and temperature measurement of virtualization solutions. In *Thermal Investigations of ICs and Systems (THERMINIC), 2011 17th International Workshop on*, pages 1–6. IEEE, 2011.

[8] D. Snowdon, E. Le Sueur, S. Petters, and G. Heiser. Koala: a platform for os-level power management. In *Proceedings of the 4th ACM European conference on Computer systems*, pages 289–302. ACM, 2009.

[9] M. Verma, S. Steinke, and P. Marwedel. Data partitioning for maximal scratchpad usage. In *Proceedings of the 2003 Asia and South Pacific Design Automation Conference*, ASP-DAC '03, pages 77–83, New York, NY, USA, 2003. ACM.

[10] D. Versick, E. Semenov, and D. Tavangarian. Virtualization techniques for energy-efficient data centers. In *Proceedings of the International Conference on Computer Science and Information Technology*, 2011.

[11] Website. Webservices for devices. http://ws4d.e-technik.uni-rostock.de/, Mar. 2012.