

An Energy Prediction Model for Cloud Data Centers Through Performance Counter

Sa Meng^{a,*}, Peng Sun^a, Jie Luo^{b,c}, and Han Xu^a

^a*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China*

^b*School of Computer Science and Engineering, Beihang University, Beijing, 100191, China*

^c*State Key Laboratory of Software Development Environment, Beihang University, Beijing, 100191, China*

Abstract

In recent years, with an increased environmental protection concern, the carbon footprints of large-scale cloud data centers have come into public view. Energy efficiency has become a key indicator for these data centers. Management personnel in cloud data centers should know the relationship between the workload patterns and the energy consumption of the infrastructure in order to optimize energy efficiency. In this paper, we first discuss the energy consumption problems of cloud data centers and summarize the related work. Then, we purpose an energy prediction model to estimate the energy consumption of servers in cloud data centers based on performance counters of their processors. Afterward, the proposed model is tested and analyzed under a wide selection of benchmarks, including SPEC2006, I/Ozone, and Netperf. Finally, through analyzing the results of the contrast experiments, it is shown that the proposed energy prediction model can predict energy consumption of cloud servers with high accuracy.

Keywords: energy prediction; accuracy; performance counter; multivariate linear regression

(Submitted on June 12, 2019; Revised on August 15, 2019; Accepted on October 20, 2019)

© 2019 Totem Publisher, Inc. All rights reserved.

1. Introduction

With the development of cloud computing, the infrastructure scale of a cloud data center is much larger than that of a traditional data center. Along with service model upgrading and increased software needs, cloud data center power consumption has been in a prominent position. Take Google's cloud data center in Oregon for example; its energy consumption is equal to the total amount consumed by all families in a medium-sized city when it operated at full load [1]. Hence, the energy consumption of cloud computing has attracted wide attention from academia and industry.

S. M. Rivoire [2] stated that physical power measurements alone cannot meet these criteria since they can neither predict future energy consumption nor provide a link between resource usage and energy consumption, which is necessary for energy-efficiency optimizations. Studies of the power pattern make great contributions in letting us know the distinct effect of individual components on energy consumption. Additionally, enhancing the control of component adjustment can minimize the energy consumption without the loss of performance. It also provides a key to optimizing the migrate algorithm.

With the advances in manufacturing technology, CPU performance is rising rapidly. Especially in the last decade, multi-core technologies are widely used and one motherboard can accommodate more CPUs, making an astonishing growth of energy consumption. For example, in the 1990s, a traditional data center rack may have a 20 CPU core, but the current figure may be thousands. However, with the increasing attention on the power pattern study, there are more self-adjust schemes or manual ones. Most existing schemes are provided by hardware and OS providers, such as Intel's Speed Step, AMD's PowerNow, CVS, and DVS, which indeed result in the complexity of the model.

* Corresponding author.

E-mail address: summerincuit@gmail.com

Although the widely used Intel processors with Sandy Bridge architecture, such as Intel® Core™ i7-2xxx, Intel® Core™ i5-2xxx, and Intel® Core™ i3-2xxx processor series, have RAPL counters for direct power reading, most processors do not provide such an interface for power measurement. Our aim is to estimate the energy consumption of the machines without RAPL counters by using the performance counter. The proposed model is based on the multivariate linear regression and reveals how the usage behavior affects the computer's energy consumption. In our experiment, the model can predict the energy consumption with high accuracy.

The remainder of this paper is organized as follows: Section 2 discusses the related works in the domain of energy consumption modeling. Then, we describe the methodology of our work in Section 3, including the performance events for training the model, the characterizations of benchmarks we selected, and the mathematical details of the proposed model. Section 4 discusses the energy prediction model and the prediction results. Section 5 compares our model with an existing widely used energy model and points out that the black-box model could be more suitable for management and maintenance of hardware in cloud data centers. Finally, in Section 6, we give conclusions about our work and discuss the possible mathematical processing methods for the energy efficiency of cloud data centers.

2. Related Work

The energy model for servers in cloud data centers has gained increasing attention. There are various approaches aimed at estimating and predicting the power consumption of cloud servers.

Fan et al. [3] showed that full-system models based on the OS-reported CPU utilizations alone proved to be highly accurate. Chung-Hsing et al. [4] studied the recent servers and stated that there were more effective power-management schemes provided by computer hardware providers or operating system providers in general, especial since 2008. They identified an exponent function appropriate for modeling the power behaviors of modern computers. However, the value of the parameter was derived empirically.

Another proposed method developed linear models using CPU, memory, disk, net utilization, board and so on. Economou et al. [5] presented a non-intrusive method to model full-system energy consumption on two server systems with different power characterizations. Lewis et al. [6] introduced thermal as a new factor besides hardware components.

Contreras et al. [7] and Singh et al. [8] introduced a linear energy model that can predict the full-system power consumption with a certain process. However, these methods need to determine the specific events before sampling, which means this method was only available for the determined architecture. Isci [9] demonstrated a processor power modeling approach that estimated the maximum power of each component and combined them together.

Some researches focused on the energy consumption analysis under special conditions. Li et al. [10] analyzed the workload data from the components of the three processing stages for MapReduce clusters, such as reduce file bytes written, and built a power prediction model using multivariate linear regression. Their data came from an application with a high CPU utilization or ones with heavy disk I/O and moderate CPU utilization. Wu et al. [11-12] placed an emphasis upon the performance characteristics of hybrid MPI/OpenMP implementations on large-scale multicore clusters. They built application-centric models for the energy consumption of the system, CPU, and memory components. They compared the different energy patterns between OpenMP and MPI, rather than the accuracy of the model building. Poess et al. [13] made an energy consumption analysis of TPC-C results, which is the transaction processing of the benchmark, and introduced a prediction model to show the system performance and power performance trends.

3. Methodology for Building an Energy Prediction Model

In order to estimate the relationship between energy consumption and resource usage, we developed the energy model using the performance events that could reflect the processing of the applications. The energy consumptions are captured by a power meter. We recorded the trace of the performance counters during the execution period when the benchmark was running. Not all of the performance events are actually remarkable indicators of the energy consumption of the machine. The performance events can be selected by the multivariate linear regression model based on a least-squares fitting algorithm in order to minimize the number of introduced events and the complexity of models without losing accuracy. Figure 1 shows the flow chart of the energy model processing steps.

3.1. Energy Estimation using Performance Counter Events

Intel processors such as Intel Core Solo and Intel Core Duo processors with architectural performance monitoring, provide a

mechanism for programs to enumerate performance events and provide configuration and counting facilities for events [14].

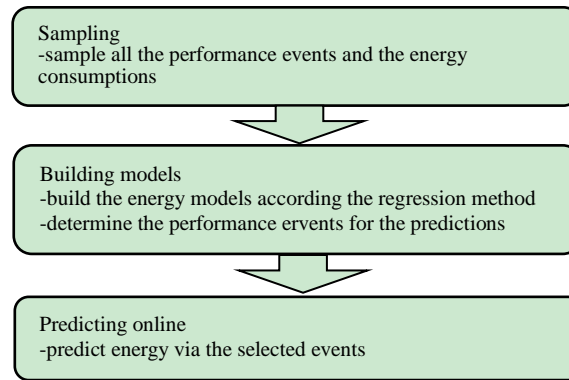


Figure 1. Flow chart of the energy model processing steps

We present the model developed from the Intel® Core™2 Duo E6550 processor. The L1 cache is divided into two sections: one section is dedicated to caching instructions and the other to caching data. The L2 cache is a unified data and instruction cache located on the processor chip, which is shared between two processor cores. Table 1 shows the sampling events that are introduced in the first step.

Table 1. List of monitoring events

	Cache instructions	Cache data
1	cpu-cycles	instructions
2	cache-references	cache-misses
3	branch-instructions	branch-misses
4	bus-cycles	task-clock
5	L1-dcache-loads	L1-dcache-load-misses
6	L1-dcache-stores	L1-dcache-store-misses
7	L1-dcache-prefetches	
8	L1-icache-loads	L1-icache-load-misses
9	LLC-loads	LLC-load-misses
10	LLC-stores	LLC-store-misses
11	dTLB-loads	dTLB-load-misses
12	dTLB-stores	dTLB-store-misses
13	iTLB-loads	
14	branch-loads	branch-load-misses

Instructions: Comparing energy consumption and performance monitoring events, it is obvious that instructions-per-cycle always makes great contributions to energy consumption. When an instruction is executed, more computing units are working; this results in higher energy consumption.

L1/LLC-Cache: Cache is used for reducing the latency of memory access and improving the data processing speed of a CPU. In general, instruction/data cache misses mean the computing units have to stall until the resources become available. Low miss or high hit of L1/LLC-Cache is beneficial to the speed execution of the instructions, thus affecting energy consumption.

TLB: The function of the translation look-aside buffer (TLB) is to map virtual addresses to physical addresses. If the TLB misses, the processor has to handle the page fault exception. Hence, the TLB misses indicates the processor can't start until the memory accomplishes the works, which affects the energy consumption.

Therefore, these performance events are monitored as influential factors.

3.2. Characterizations of Benchmarks

In order to get the full comprehension of the server, we use a wide selection of benchmarks, including SPEC2006 [15], I/Ozone [16] and Netperf [17]. All of these stand for the various workloads with distinguished features in the clusters.

SPEC2006 often requires extreme CPU capacity with low disk I/O access, while I/Ozone and Netperf demand more resources on the disk throughput and network bandwidth, respectively.

Through monitoring the system resources usage, these benchmarks show quite different footprints and characterizations in terms of including CPU usage, memory load, and network traffic, which are compared in Figures 2 to 4.

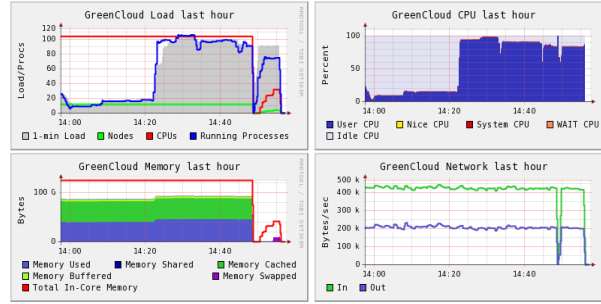


Figure 2. System resource utilization on cluster of SPEC2006

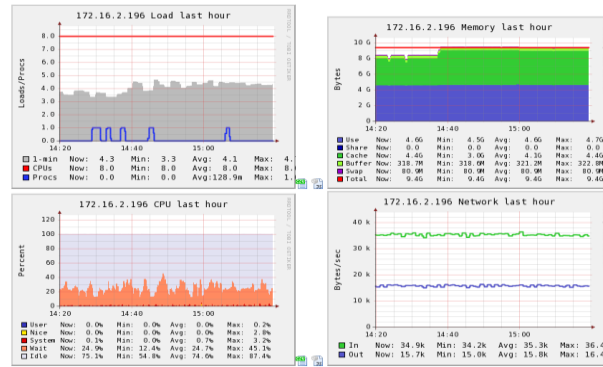


Figure 3. System resource utilization on one node of I/Ozone

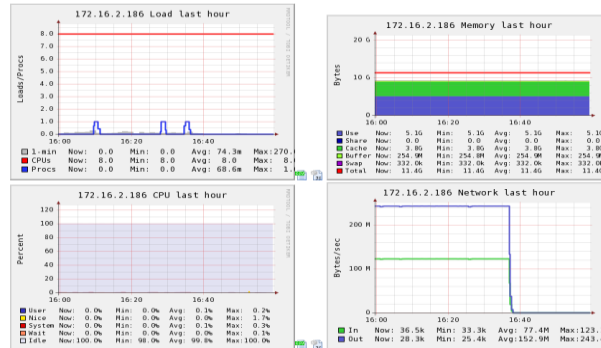


Figure 4. System resource utilization on one node of Netperf

3.3. Stimulated Energy Prediction Models

3.3.1. Multiple Linear Regression Model

Multiple linear regression is a statistical method for modeling the linear relationship between a scalar variable y and m , with explanatory variables denoted $X_1, X_2, X_3, \dots, X_n$. It can be written as:

$$\begin{cases} y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_m x_{im} + \varepsilon_i \\ E\varepsilon_i = 0, D\varepsilon_i = \sigma^2 \end{cases} \quad (1)$$

Where $\beta_0, \beta_1, \beta_2, \dots, \beta_m$ are regression factors to be estimated, independent of x_1, x_2, \dots, x_m .

Simply, it can be taken by matrix:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1m} \\ 1 & x_{21} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{pmatrix} = (1_n, X_\beta), \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}$$

Equation (1) can be rewritten in the matrix form:

$$\begin{cases} y = X\beta + \varepsilon \\ E(\varepsilon) = 0, D(\varepsilon) = \sigma^2 I_n \end{cases} \quad (2)$$

In this linear model, $\beta_0, \beta_1, \beta_2, \dots, \beta_m$ is determined according to the least Squares Estimator, minimizing the objective function:

$$\begin{aligned} Q(\beta_0, \beta_1, \dots, \beta_m) &= \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \cdots - \beta_m x_{im})^2 \\ &= (y - X\beta)^T (y - X\beta) \end{aligned} \quad (3)$$

Then, the fitted linear regression is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m \quad (4)$$

3.3.2. Stepwise Regression

The stepwise Regression method is used to choose the explanatory variables in an automatic procedure. In general, there are three methods:

(1) Forward selection: at the beginning, there are no variables in the model. The variables are chosen in the model if it is statistically significant. Stop the procedure until no variables can be included in the model.

(2) Backward selection: including all potential explanatory variables in the model, each step eliminates variables that improve the model until none can be excluded.

(3) Bidirectional elimination: a combination of both methods.

In this paper, we used bidirectional elimination methods to build the energy model.

4. Experiments

4.1. Hard and Software Environment

The experiment servers have a Intel(R) Core(TM)2 Duo E6550 CPU with 2048MB main memory and runs the Linux OS with patched kernel 2.6.32-220.el6.x86_64. The tool we used to monitor the performance event is perf 2.6 and the version of the compiler is gcc-4.4.6.

4.2. Energy Prediction Models

After running the extensive experiment of the benchmarks, we collect the dataset including the measurement of energy consumption and the values of all the performance counters. Then, we divided the dataset into two groups: one for training the linear model following the step regression method, the other for validating the prediction model.

According to the method we discussed above, the fitted multiple linear regression model is shown in Equation (5).

$$Y = C + \sum_{i=1}^5 \beta_i X_i \quad (5)$$

Table 2. Multiple linear regression model

Variables	Description	Parameters
Y	energy consumption	
C	constants	95.853
X_1	task-clock	24.646
X_2	L1-dcache-prefetches	0.375
X_3	LLC-load-misses	0.227
X_4	dTLB-store-misses	-1.143
X_5	dTLB-stores	-0.011

As shown in Table 2, the performance counters, including Task- clock, L1-dcache-prefetches, and LLC- load- misses, show a positive correlation with energy consumption, while dTLB and other events indicate that the memory even disk accesses have a negative correlation with energy consumption. Hence, the processor is still the main component of energy consumption in our server.

Figure 5 shows the comparisons between the estimations of energy consumption and observations. In addition, Figure 6 is the error scatter plot of the residuals versus the measured values of the benchmark applications in the training cases.

$$\varepsilon = \frac{Power_{predict} - Power_{measure}}{Power_{measure}} \quad (6)$$

The error plot shows a random pattern, indicating that the residual is independent of the fitted values and the model fit normality assumptions. Statistically, most of the prediction cases fall into the (-3%, 3%) error range, which states that the energy consumption model based on performance counters is very reliable.

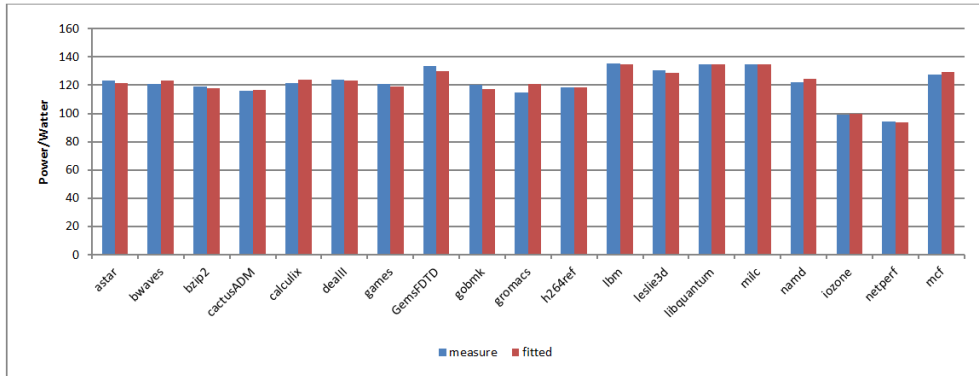


Figure 5. Estimations and observations of training cases

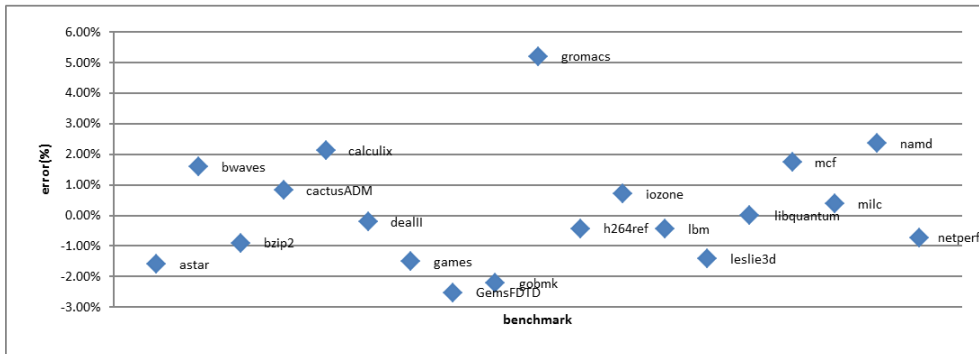


Figure 6. Error plot of training cases of the energy prediction model

4.3. Energy Predictions

In order to cross-validate our prediction model, we selected the other applications in SPEC2006 to predict their powers and check the model accuracy according to Equation (5). Figure 7 displays the bar chart of the comparisons between the estimations of energy consumption and the observation ones.

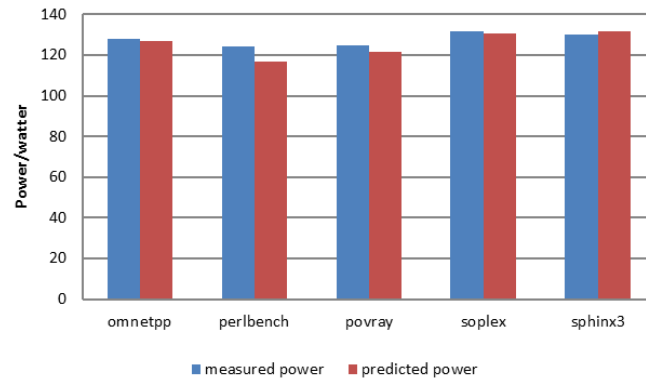


Figure 7. Estimations and observations

Figure 8 shows that most of the prediction errors fall into the (-3%, 3%) range, and the maximum values of the errors are not more than 6 percent. Consequently, this model can predict the energy consumption with high accuracy.

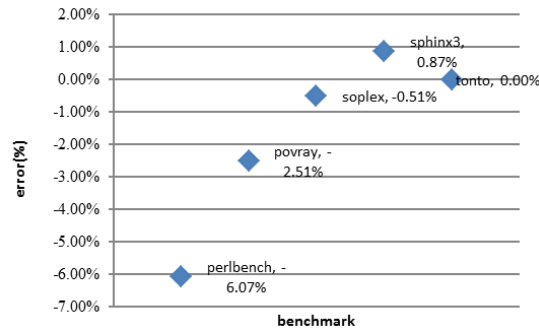


Figure 8. Error plot of predictions

5. Comparisons Between Our Model and an Existing Model

To illustrate the accuracy of the proposed model, we compare it with an existing model in which the performance events in the models are all determined before monitoring [7]. In addition, we constructed the model by using the performance events introduced in Table 3.

$$Y = C + \sum_{i=1}^5 \beta_i X_i \quad (7)$$

Table 3. Comparison of model parameters

Variables	Description	Parameters
Y	energy consumption	
C	constants	-8.401
X_1	instructions	60.888
X_2	L1dcacheprefetches	0.552
X_3	L1icacheloads	0.054
X_4	dTLBstoremisses	2.511
X_5	iTLBloads	-0.026

Figures 9-12 show that most of the prediction cases fall into the (-15%, 15%) error range, which is much higher than our build above.

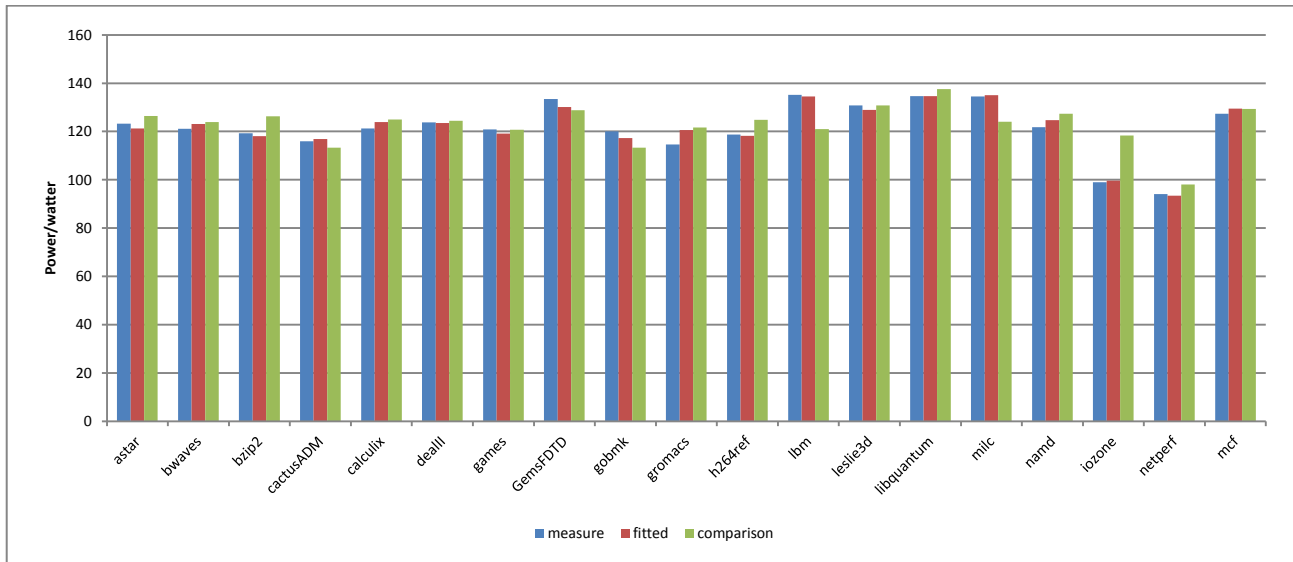


Figure 9. Estimations and observations of training cases

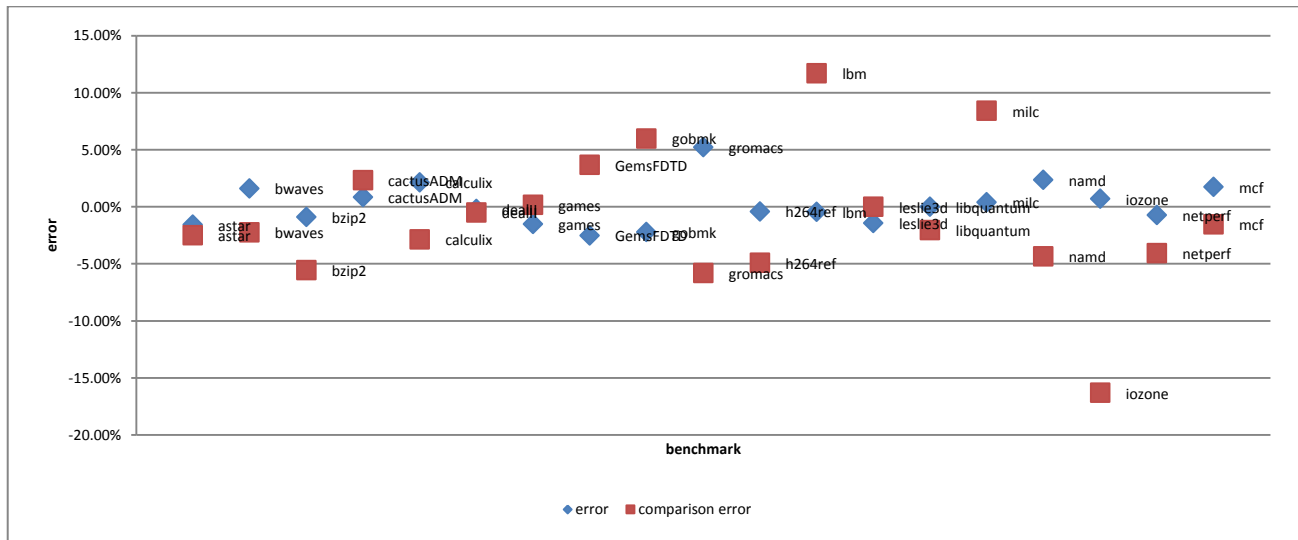


Figure 10. Error plot of training cases of two energy prediction models

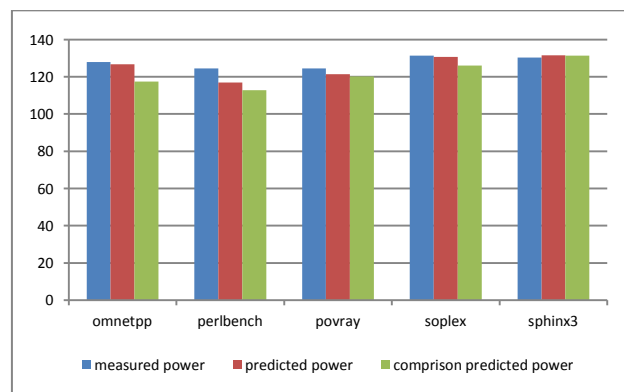


Figure 11. Estimations and observations of the energy prediction model

As discussed above, we also predicted the energy consumption of the other five cases. Although the error range is (-1%, 10%), the errors of four cases exceeded 5%. Therefore, the error plot did not fit the random pattern, implying that the model has an obvious deviation.



Figure 12. Errors of the energy prediction model

All of the results show that, if the users of the cluster don't choose the monitoring performance event seriously, the model may not predict the energy consumptions well. However, in a data center, the processors may be quite distinct and the system managers may know few details, and so, it is difficult for the managers to determine certain events before making the power prediction models. Moreover, our method can construct an energy prediction model without implementation knowledge.

6. Conclusions

Energy prediction is essential for system management in cloud data centers. In this paper, we propose a model to estimate the energy consumption of cloud servers based on the performance counter. The analysis and experiments reveal the relationship between resource usage behaviors and energy consumption. The proposed model has been tested under a wide selection of benchmarks, including SPEC2006, I/Ozone, and Netperf. The experiment demonstrated that the proposed energy prediction model can predict energy consumption with high accuracy. We used a multivariate linear regression method to construct the energy prediction model that is suitable for predicting future energy consumption through resource utilization of cloud servers.

References

1. S. Ghemawat, H. Gobioff, and S. -T. Leung, "The Google File System," in *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, NY, USA, October 2003
2. S. Rivoire, "Models and Metrics for Energy-Efficient Computer System," Ph. D. Dissertation, Stanford University, 2008
3. X. Fan, W. D. Weber, and L. Barroso, "Power Provisioning for a Warehouse-Sized Computer," in *Proceedings of International Symposium on Computer Architecture (ISCA07)*, pp. 13-23, June 2007
4. C. -H. Hsu and S. W. Poole, "Power Signature Analysis of the SPECpower_ssj2008 Benchmark," in *Proceedings of Performance Analysis of Systems and Software (ISPASS)*, pp. 227-236, April 2011
5. D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-System Power Analysis and Modeling for Server Environments," in *Proceedings of Workshop on Modeling, Benchmarking, and Simulation (MoBS)*, June 2006
6. A. Lewis, S. Ghosh, and N. -F. Tzeng, "Run-Time Energy Consumption Estimation based on Workload in Server Systems," in *Proceedings of the 2008 Conference on Power Aware Computing and Systems (HotPower'08)*, pp. 4, 2008
7. G. Contreras and M. Martonosi, "Power Prediction for Intel XScale® Processors using Performance Monitoring Unit Events," in *Proceedings of Low Power Electronics and Design (ISLPED05)*, pp. 221-226, August 2005
8. K. Singh, M. Bhaduria, and S. A. McKee, "Real Time Power Estimation and Thread Scheduling via Performance Counters," *ACM SIGARCH Computer Architecture News Archive*, Vol. 37, pp. 46-55, May 2009
9. C. Isci and M. Martonosi, "Runtime Power Monitoring in High-End Processors: Methodology and Empirical Data," in *Proceedings of the 36th International Symposium on Microarchitecture (MICOR 36)*, pp. 93-97, May 2003
10. W. Li, H. Yang, Z. Luan, and D. Qian, "Energy Prediction for MapReduce Workloads," in *Proceedings of Dependable, Autonomic and Secure Computing (DASC11)*, pp. 443-448, December 2011
11. X. Wu and V. Taylor, "Performance Characteristics of Hybrid MPI/OpenMP Implementations of NAS Parallel Benchmarks SP and BT on Large-Scale Multicore Clusters," *The Computer Journal Advance*, Vol. 55, pp. 154-167, July 2011
12. C. Lively, X. Wu, V. Taylor, S. Moore, H. -C. Chang, C. -Y. Su, et al., "Power-Aware Predictive Models of Hybrid (MPI/OpenMP) Scientific Applications on Multicore Systems", *Computer Science - Research and Development*, Vol. 27, No. 4, pp. 219-235, 2011

13. M. Poess and R. O. Nambiar, "Energy Cost, The Key Challenge of Today's Data Centers: A Power Consumption Analysis of TPC-C Results," *VLDB Endowment*, Vol. 1, pp. 1229-1240, August 2008
14. Intel® 64 and IA-32 Architectures Software Developer's Manual, Intel Corporation, 2012
15. F. Hameed, A. A. Khan, and J. Castrillon, "Performance and Energy-Efficient Design of STT-RAM Last-Level Cache," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 99, pp. 1-14, 2018
16. G. Ateniese, R. Burns, and R. Curtmola, "Remote Data Checking using Provable Data Possession," *ACM Transactions on Information and System Security*, Vol. 14, No. 1, pp. 1-34, 2011
17. A. Menon, S. Schubert, and W. Zwaenepoel, "TwinDrivers: Semi-Automatic Derivation of Fast and Safe Hypervisor Network Drivers from Guest OS Drivers," *ACM Sigarch Computer Architecture News*, Vol. 37, No. 1, pp. 301-312, 2009

Sa Meng is currently pursuing a Ph.D. at the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). Her current research interests include cloud computing, reliability modeling and optimization, and energy-efficient computing.

Peng Sun is an Assistant Professor at the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). His research interests include cloud computing, reliability modeling, and optimization.

Han Xu is currently pursuing a Ph.D. at the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). His research interests include system optimization of reliability, performance, power consumption and safety.