Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2015, Article ID 143071, 9 pages
http://dx.doi.org/10.1155/2015/143071

Research Article

Regression Cloud Models and Their Applications in Energy Consumption of Data Center

Yanshuang Zhou, Na Li, Hong Li, and Yongqiang Zhang

Handan College, Handan, Hebei 056005, China

Correspondence should be addressed to Yanshuang Zhou; shuangshuang223@126.com

Received 13 August 2015; Accepted 27 September 2015

Academic Editor: James Nightingale

As cloud data center consumes more and more energy, both researchers and engineers aim to minimize energy consumption while keeping its services available. A good energy model can reflect the relationships between running tasks and the energy consumed by hardware and can be further used to schedule tasks for saving energy. In this paper, we analyzed linear and nonlinear regression energy model based on performance counters and system utilization and proposed a support vector regression energy model. For performance counters, we gave a general linear regression framework and compared three linear regression models. For system utilization, we compared our support vector regression model with linear regression and three nonlinear regression models. The experiments show that linear regression model is good enough to model performance counters, nonlinear regression is better than linear regression model for modeling system utilization, and support vector regression model is better than polynomial and exponential regression models.

1. Introduction

In a 7 × 24 running cloud data center, energy model is one of the most important researches in green computing [1]. In a cloud data center, or data center for short, both users and administrators need to know how the energy of servers is consumed according to their behaviors, and thus they can minimize the total consumed energy. Along with support of energy monitoring by hardware, energy consumption of most hardware can be measured in run-time [2]. However, in order to further predict future energy consumption, researchers need to build energy model by their own. During the modeling of energy consumption, researchers need to capture the relationships between hardware resources and their energy consumptions.

Nowadays, as the scale of data center becomes larger and larger, the total energy consumption of data center becomes bigger and bigger and the energy model becomes more complex. Each cloud service, such as SaaS (Software as a Service) [3], PaaS (Platform as a Service) [4], or IaaS (Infrastructure as a Service) [5], has different energy optimization methods, and these methods cannot be migrated to other services directly. However, a unified energy framework [6] includes the following three basic steps. Firstly, it needs to sample or collect factors that affect the energy consumption of the whole system or some hardware components. Commonly used sampling methods are classified into performance counters based and system utilization based. After collecting factors that affect energy consumption, we must combine them together with a suitable model. The more suitable the model, the more accurate the energy prediction we will get in the future. Finally, based on the proposed model, one can focus on factors that affect energy consumption the most, monitor their energy consumptions, and schedule running tasks to minimize the total energy consumption.

In this paper, we analyzed the energy consumption of data center by each computing node and analyzed different regression models and their performances on predicting energy consumption. The contributions of the paper are as follows:

(i) We gave a performance counter based energy model framework and analyzed different linear regression models. According to experiments, we concluded that all kinds of linear models had similar prediction performance, and it was enough to model energy consumption with linear models for the performance counters.

(ii) We analyzed linear and nonlinear regression models based on system utilization and proposed a support vector regression model.

(iii) We did lots of experiments for the system utilization based energy models and had the following conclusion. Nonlinear regression models are better than linear regression model, and our support vector regression model is better than polynomial and exponential regression models.

The rest of the paper is organized as follows. Section 2 describes related works about energy model. In Section 3, we give a performance counter based energy model framework, analyze different regression methods for the system utilization based energy model, and propose a support vector regression energy model. Experiments and conclusion are in Sections 4 and 5, respectively.

2. Related Works

In order to control or reduce the energy consumption of a huge data center, researchers should first understand how energy is consumed, so building energy model for a data center is the basis. Most energy models can be classified into system utilization based and performance counters based [6]. However, there are also some environment oriented models.

2.1. System Utilization Based Model. In system utilization based model, researchers model the energy consumption of the whole system based on components. Fan et al. [7] built a linear regression model for total energy consumption of the whole system using CPU utilization and validated its effectiveness. However, after studying the 177 specimen of SPECpower from years between 2007 and 2010, Hsu and Poole [8] found that energy consumption of servers had changed a lot. Since 2008, both hardware providers and system software providers had proposed their own energy management frameworks, and many of them are not simple linear models. In [8], Hsu and Poole proposed a unitary exponential model for computation intensive applications, and the parameters in the model were from experience, and thus their model were not applicable to other applications. Beloglazov et al. [9] utilized the maximal power of servers, CPU utilization, and the energy ratio of idleness to maximal power to model the energy consumption.

In addition, there are some researches that study the whole system by analyzing each component. These researches detached the whole system into CPU, memory, disks, I/O, and so on and built multiple linear regression models. Economou et al. [10] applied noninvasive method to model energy consumption of the whole system, and the model was built after one calibration. In the experiments, they compared the performance of their model on servers with different energy and workload. Beside hardware components, Lewis et al. [11] took the temperature of environment and bus activities into consideration and predicted the long time energy consumption of servers. Kliazovich et al. [12] modeled the energy of servers by power and frequency of CPUs and a constant. In their model, all components except for CPUs were considered to be constant. In the DENS schedule algorithm, Kliazovich et al. [13] described an energy model of the whole data center, where computing nodes were linear model of CPU utilization, and switch was multivariate function. Lee and Zomaya [14] used the maximal power, the minimal power, and current power of CPUs to model the energy consumption.

2.2. Performance Counter Based Model. In order to further improve the accuracy of energy model, researchers proposed performance counter based energy models along with the development of CPU. By reading data from performance counters together with features of the underlined components, many energy models could be built. For example, according to analyzing memory energy consumption, researchers found that the main energy consumption for memory was read/write throughput [8]. In order to acquire throughput of memory, we can record the missing number of the last level cache, and this can be easily accessed.

In addition, some researchers used data of performance event to build energy model. They studied specified CPUs and fixed the events that needed to be sampled; however their methods could not be applicable to other situations. Isci and Martonosi [15] sampled data for 22 events and built linear energy model based on all events' maximal power. This model is much more complex and needs to sample much data and cannot be deployed into cloud data center.

2.3. Environment Oriented Model. For some specified computing environments, researchers studied environment oriented energy models. In the MapReduce environment, Li et al. [16] analyzed the workload data for each Map, Merge, and Reduce step and built an energy model based on linear regression. They analyzed computing intensive and I/O intensive two classical MapReduce applications. For MPI/OpenMP environment in large scale multicore data center, Wu et al. [17, 18] built an application-centered energy model for CPUs and memory components, and the main target of their research is to differentiate the energy consumption between OpenMP and MPI.

In addition, Farahnakian et al. [19] propose a KNN regression approach for predicting energy consumption in data centers. However, this approach is a heuristic regression model, but we analyze different kinds of regression models, especially on support vector regression. Beloglazov et al. [9] also propose a heuristic energy predicting approach and the underlying model is not a regression model. By experiments, we validate that regression model is good enough for prediction in such situation, so nonregression model is not necessary.

3. Modeling Energy with Regression Methods

3.1. Modeling Performance Counter. Since Pentium series, the Intel processors have introduced performance monitoring system by some specified performance counters. According to these performance counters, we can acquire the performance of processors and thus optimize the performance of the compiler and even the whole system. Generally speaking, the events that can be monitored in processors include instructions, cache, and table lookup buffer. Though these events affect energy consumption of the whole system, they have different weights. By linear equation, we give different events different weights and estimate these weights by historical data. The energy model based on performance counters is as follows:

$$P_c = C_0 + \sum_{i=1}^{n} C_i p_i, \tag{1}$$

where C_0 is a constant, n is the number of monitored events, p_i is the sampled event of performance counter, which represents its energy consumption, and C_i is the weight of event i.

Nowadays, these are many tools that can be used to collect information of performance counters, such as OProfile [20] and Perf [21]. By these tools, applications can acquire energy statistics of hardware components with power management unit. Perf is a system performance optimization tool in Linux kernel and has advantages in adaptation and developing activity. In this paper, we apply Perf to sample performance counters.

3.2. Modeling System Utilization. Except for performance counters, we also apply system utilization to model the energy consumption. In this paper, we mainly focus on system utilization of CPU, memory, storage, and I/O and consider other parts of the system as constant.

3.2.1. CPU Utilization Energy Model. CPU consumes lots of energy in a system, but it can adjust energy consumption conveniently. So adjusting the status of CPUs in a data center is the main energy-saving method.

In [22], the energy model of CPUs is a simple linear regression, which can be described as

$$P = C_0 + C_1 U_{\text{CPU}}, \tag{2}$$

where C_0 is a constant, U_{CPU} is the utilization of CPU, and C_1 is its weight. However, the energy consumption of CPU is not linear with its utilization, so we add another empirical parameter to modify their relation:

$$P = C_0 + C_1 U_{\text{CPU}} + C_2 U_{\text{CPU}}^{t}, \tag{3}$$

where U_{CPU}^{t} is a calibrating parameter and C_2 is its weight.

3.2.2. Memory Utilization Energy Model. When memory is idle, we can make it standby, sleep, or even shutdown to save the energy consumption. In each of the above statuses, the energy consumption is lower than working status but the response time increases. According to our observation, the energy consumption of memory is proportional to CPU. That is, when CPU utilization is high, the energy consumption of memory is also high. So, we model the energy consumption of memory as follows:

$$P = C_0 + C_1 U_{\text{CPU}} + C_2 U_{\text{memory}}, \tag{4}$$

where U_{memory} is the utilization of memory and C_2 is its weight.

3.2.3. Storage Utilization Energy Model. Because of the scalability of server, there are usually more than one disk in a server, for example, multiple disks or disk arrays. Different storage systems have different energy consumption. The method for saving energy consumption of storage system is to apply disks with different adjusting functions, such as changing from running status to standby or sleep status, or adjusting the revolutions per minute. In this paper, we consider that the energy consumption of storage is also related to CPU and the equation is

$$P = C_0 + C_1 U_{\text{CPU}} + C_2 U_{\text{disk}}, \tag{5}$$

where U_{disk} is the utilization of storage system and C_2 is its weight.

3.2.4. I/O Utilization Energy Model. We also observed the energy consumption of I/O system. In a server, the I/O energy consumption mainly depends on network interface card and data transition. In this paper, we study the energy model of data center by nodes, and the energy used by data transition between switches is not considered. The energy model of I/O system is as follows:

$$P = C_0 + C_1 U_{\text{CPU}} + C_2 U_{\text{IO}}, \tag{6}$$

where U_{IO} is the utilization of I/O system and C_2 is its weight.

3.3. Regression Methods. Regression method is an effective tool for modeling the relationship between variables, and its purpose is to predict the change of target variable with respect to changes of decision variables. Concretely speaking, we assume energy as the target variable y and utilizations of components as decision vector $\mathbf{x}$. Given some pairs of $\mathbf{x}$ and y, we seek to find the relationship between them; that is, $y = \mu(\mathbf{x})$, and this method is called regression analysis.

3.3.1. Linear Method. In regression analysis, the target variable is called dependant variable, and the decision variables are called independent variables. Multivariate linear regression models the relationship between dependant variable and multiple independent variables. If there are m independent variables, then the regression model is

$$\begin{gathered} y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \varepsilon, \\ E(\varepsilon) = 0, \\ D(\varepsilon) = \sigma^2 < +\infty, \end{gathered} \tag{7}$$

where y is the observed energy consumption, x_i $(1 \le i \le m)$ is the observation of utilization of each component, β_i $(0 \le i \le m)$ is the coefficient of regression model, and ε is the random error that cannot be observed. For random error ε, we assume that its expectation and variance are 0 and σ^2, respectively.

If there are n observation samples $(y_i, x_{i1}, \ldots, x_{im})$, $i = 1, \ldots, n$, then

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_m x_{im} + \varepsilon_i, \\ E(\varepsilon_i) &= 0, \\ D(\varepsilon_i) &= \sigma^2, \end{aligned} \tag{8}$$

where ε_i $(i = 1, \ldots, n)$ are independent of each other. For convenience, we represent the above equation with matrix and vector, and then we get

$$\begin{aligned} y &= X\beta + \beta_0 + \varepsilon, \\ E(\varepsilon) &= 0, \\ D(\varepsilon) &= \sigma^2 I, \end{aligned} \tag{9}$$

where

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}$$

$$y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix},$$

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}, \tag{10}$$

$$\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{pmatrix}.$$

Next, we train the above linear regression model with sampled system utilization. We apply least squares method to estimate the regression coefficient, and the objective is minimizing the following equation:

$$\begin{aligned} &Q(\beta_0, \beta_1, \ldots, \beta_m) \\ &\quad = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_m x_{im})^2 \\ &\quad = (y - \beta_0 - X\beta)^T (y - \beta_0 - X\beta). \end{aligned} \tag{11}$$

When we minimize the above equation at $\beta'_0, \beta'_1, \ldots, \beta'_m$ and y', we get the following regression model:

$$y' = \beta'_0 + \beta'_1 x_1 + \cdots + \beta'_m x_m. \tag{12}$$

Generally speaking, it is very hard to get the optimal solution using the least squares method, so we apply the one-by-one regression method. The main idea is introducing one variable at each time, and the condition when introducing a variable is that the bias of the variable is significant. At the same time, after introducing a variable, we check all introduced variables and delete nonsignificant variables. When we have checked all variables, we get the final regression model.

With the one-by-one regression method, we get noncontinuous parameters when normalized. In order to get continuous normalized parameter, we apply lasso regression method after that. The lasso regression method uses absolute value of model parameters to penalize those bigger parameters, and the equation is

$$\begin{aligned} \min_{(\beta_0, \beta)} \Bigg[&\frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_m x_{im})^2 \\ &+ \lambda \|\beta\|_1 \Bigg], \end{aligned} \tag{13}$$

where $\|\beta\|_1 = (1/m)\sum_{i=1}^{m} \beta_i$ is the lasso regression and λ is its weight. Compared with traditional model selection method, the lasso method can prohibit overfitting of the selected model. Based on lasso regression in (13), we can make unimportant parameters to 0. When we train the model with system utilization data, we can get a balance between overfitting and underfitting. There is one thing needed to be considered; that is, the selection of λ needs cross-validation.

3.3.2. Nonlinear Methods. Based on linear model, we also introduce nonlinear regression model. By transforming function and adding factors, we get the following nonlinear regression model:

$$f(x) = w \cdot \phi(x) + b. \tag{14}$$

In (14), the transforming function can be of any form. In this paper, we apply the following three functions.

(i) Polynomial + Lasso. Instead of linear model, we use polynomial to model the relationship between system utilization and energy consumption. Here, we set $\phi(x) = \{x_i^a : 1 \le a \le 3,\ 1 \le i \le m\}$, and its coefficient is w. With lasso constraint on $\phi(x)$, the final objective is

$$f(x) = w \cdot \phi(x) + b + \lambda \|\phi(x)\|_1. \tag{15}$$

(ii) Polynomial + Exponential + Lasso. Based on polynomial and lasso regression, we add exponential transform for each coefficient and get the following equation:

$$f(x) = w \cdot \exp(\phi(x)) + b + \lambda \|\phi(x)\|_1, \tag{16}$$

where exponential function $\exp(\cdot)$ is elementwise.

(iii) Support Vector Regression. We also implement a support vector machine based regression model. Given a collection of samples (x_i, y_i), $1 \le i \le n$, where $x_i \in R^m$ are independent

variables, $y_i \in R$ is dependent variable, and n is the number of samples, the regression model is

$$f(x) = w \cdot x + b, \tag{17}$$

and the optimization problem becomes

$$\begin{gathered} \min_{w,b} \frac{1}{2} \|w\|^2, \\ \left| y_i - w \cdot x_i - b \right| \le \varepsilon, \quad i = 1, \ldots, n. \end{gathered} \tag{18}$$

For solving the above optimization problem, one can transform the original problem into its dual form, and details can be found in [23].

3.3.3. Evaluation Metric. In this paper, we use relative error $\text{Power}_{\text{error}}$ to evaluate the accuracy of the energy model, and its computation is in the following equation:

$$\text{Power}_{\text{error}} = \frac{\text{Power}_{\text{prediction}} - \text{Power}_{\text{real}}}{\text{Power}_{\text{real}}}, \tag{19}$$

where $\text{Power}_{\text{prediction}}$ is the result that we get using our model and $\text{Power}_{\text{real}}$ is the real energy consumption that we measure using electricity meter.

4. Experiments

4.1. Results on Performance Counter Based Model

4.1.1. Experimental Setup. In this experiment, the node of the server contains a Interl Xeon E5620 processor and 48 GB memory; the kernel of the Linux is 2.6.32-220.el6.x86_64. The tool we use to sample performance counters is perf-2.6.32, and the compiler is gcc-4.4.6.

4.1.2. Dataset. In order to sample data efficiently, we use SPEC2006 as our benchmark. The SPEC2006 contains integer and float operations, and we use them separately. For each operation, we use some of data as training data and the remaining as test data. We analyze the effectiveness of different mathematical methods of performance counters to energy consumption and the experimental data is in Table 1.

4.1.3. Sampling and Modeling. We run each Benchmark ten times on the training data and sample the following events: cpu-cycles, L1-dcache-loads, LLC-loads, stalled-cycles-frontend, L1-dcache-load-misses, LLC-load-misses, stalled-cycles-backend, L1-dcache-stores, LLC-stores, instructions, L1-dcache-store-misses, LLC-store-misses, stalled, cycles, per, insn, L1-dcache-prefetches, LLC-prefetches, cache-references, L1-dcache-prefetch-misses, LLC-prefetch-misses, cache-misses, L1-icache-loads, context-switches, branch-instructions, L1-icache-load-misses, cpu-migrations, branch-misses, dTLB-loads, alignment-faults, bus-cycles, dTLB-load-misses, emulation-faults, task-clock, dTLB-stores, branch-loads, page-faults, dTLB-store-misses, branch-load-misses, minor-faults, iTLB-loads, major-faults, and iTLB-load-misses.

Table 1: Experimental data.

Benchmark	Training data	Test data
Integer	bzip2, gcc, sjeng, h264ref, astar, xalancbmk, gobmk	hmmer, libquantum, mcf, omnetpp
Float	bwaves, gamess, zeusmp, gromac, cactusADM, calculix, GemsFDTD, tonto, wrf, sphinx3, dealII, soplex, povray	lbm, leslie3d, milc, namd

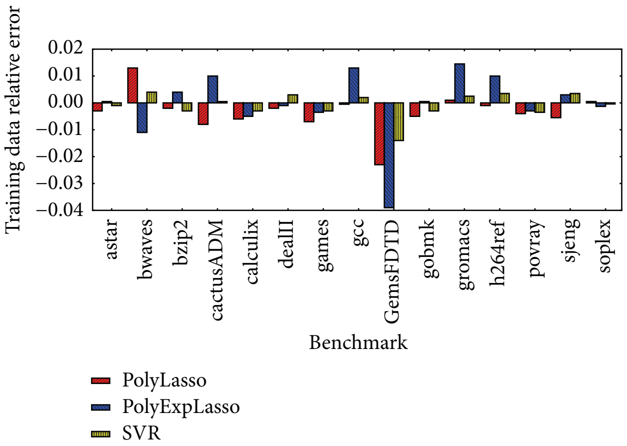

Figure 1: Relative error in training data.

After sampling data from the above events, we apply polynomial lasso regression (PolyLasso), polynomial exponential lasso (PolyExpLasso), and support vector regression (SVR), three regression models to predict the energy consumption.

4.1.4. Results. We divide the data into training data and test data. The relative error in training data reflects fitting of models to training data. In test steps, we abbreviate the real energy consumption as RealVal and compare the predicted power with its real power value.

Figure 1 illustrates the relative error in training data. As can be seen from the figure, all of the three models have small relative errors in training data and they are all between −2% and 2%. At the same time, the SVR model has better fitting advantage. Figure 2 compares the real power usage with the three predicted values on training data, and all of them are very close. Figure 3 describes the relative errors of the three models on test data, and they are all between −4% and 4%. Figure 4 compares the real power usage with the three predicted values on test data, and they are very close too.

From the above experiments we can see that the three regression models have almost the same prediction performance, and SVR does not have advantage in test data. So, in performance counter based energy model, we can apply linear regression.

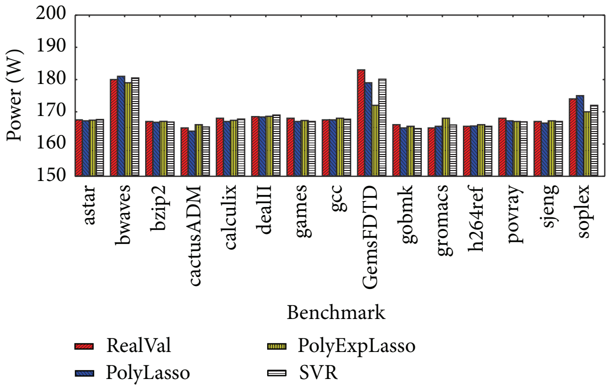

Figure 2: Result comparison on training data.

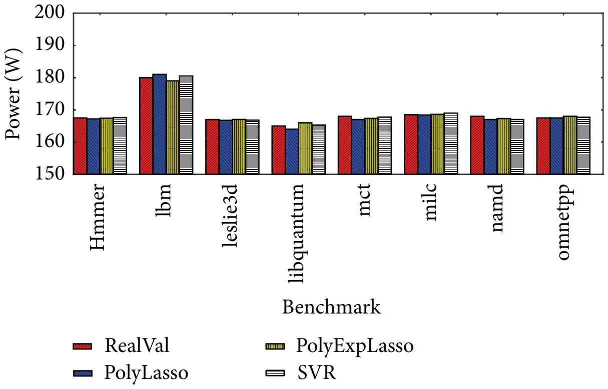

Figure 4: Result comparison on test data.

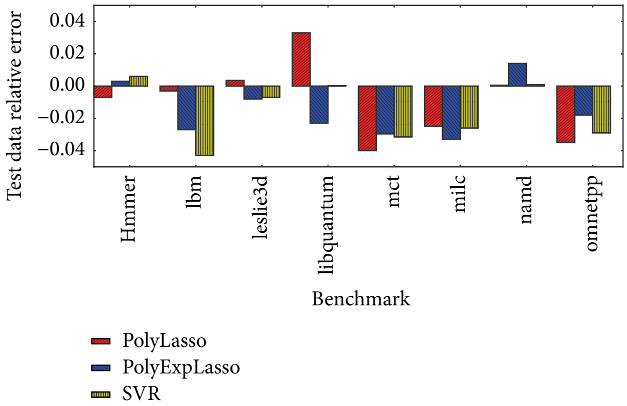

Figure 3: Relative error in test data.

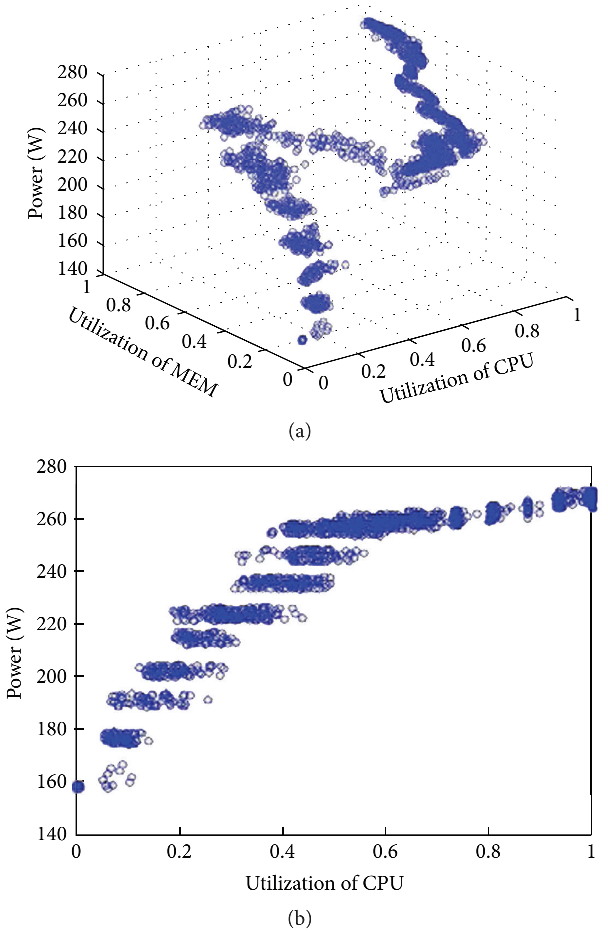

Figure 5: Energy consumption of CPU and memory utilization.

4.2. Results on System Utilization Based Model

4.2.1. Experimental Setup. The experimental environment is the same as before.

4.2.2. Sampling and Modeling. In Linux or Unix system, the utilization of processor includes user status, system status, and idle status, and they represent the statuses by which processor executes user instructions, system kernel instructions, and idle instructions, respectively. The commonly used processor utilization is the total time in which the processor executes nonidle instructions.

In this paper, the experimental operating system is CentOS6.2, and we can acquire the processor utilization with the file "/proc/stat." In this experiment, the overload of processor is as follows:

```
cpu 245679 759 89549 345763563 48744 29 2316;
cpu0 196816 395 4962 172780284 45243 0 44;
cpu1 48862 364 84587 172983278 3501 29 2271.
```

The meaning of each data is as follows:

(i) user: total time of user status since the system startup; it does not contain nice value with negative process;

(ii) nice: the total processor time of nice with negative process;

(iii) system: total time of kernel status;

(iv) idle: total waiting time except for waiting disk;

(v) Iowait: total time for waiting disk I/O;

(vi) irq: total time of hard interrupt;

(vii) softirq: total time of soft interrupt.

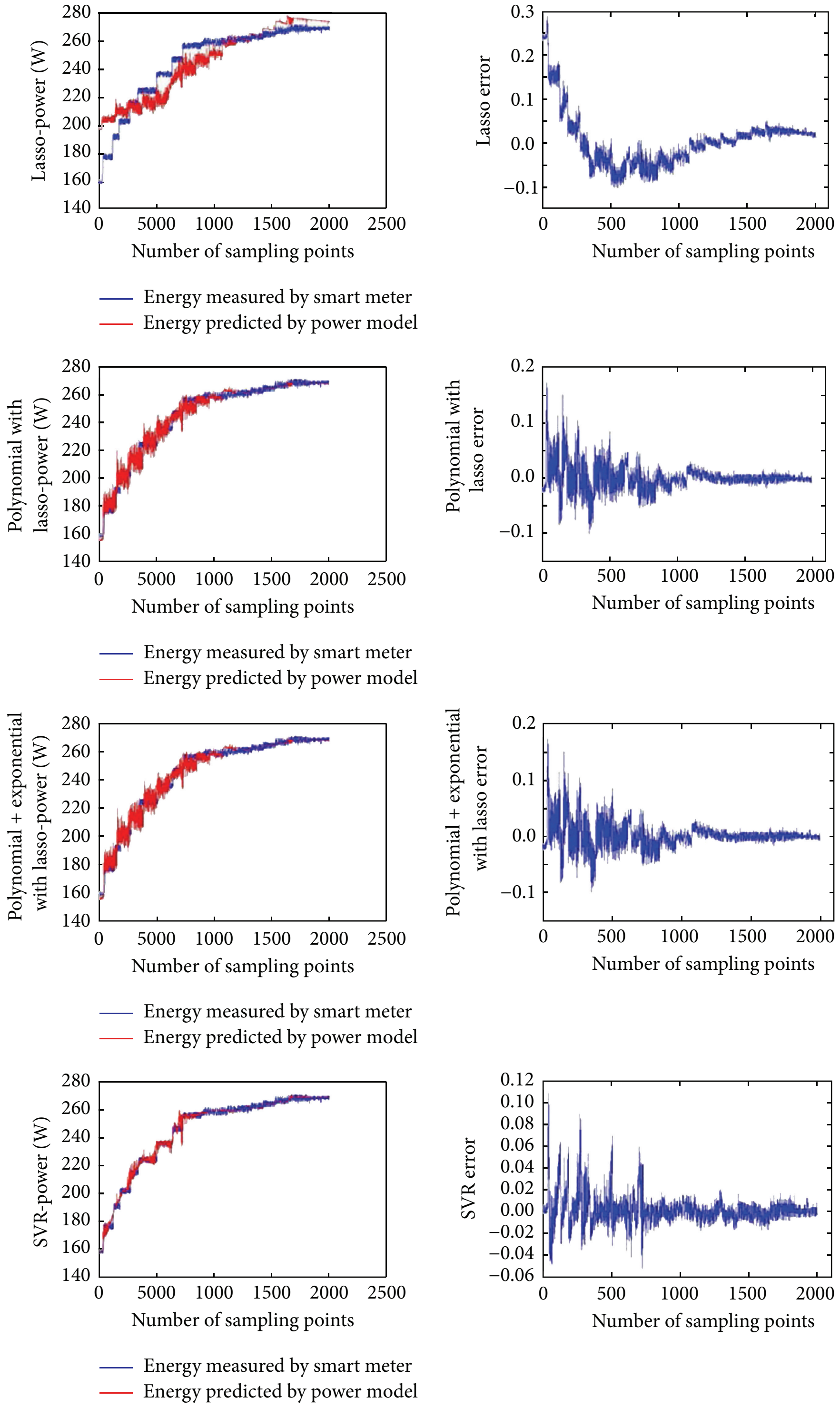

Figure 6: Results of system utilization based models.

Consider the following:

$$\text{CPU\_Time} = \text{user} + \text{system} + \text{nice} + \text{idle} + \text{iowait} + \text{irq} + \text{softirq},$$

$$t_{\text{stat}} = t_{\text{stat\_now}} - t_{\text{stat\_last}},$$

$$\text{Utilization}_{\text{CPU}} = \frac{t_{\text{user}} + t_{\text{system}}}{t_{\text{user}} + t_{\text{system}} + t_{\text{idle}}}. \tag{20}$$

With similar idea, we can acquire memory utilization with file "/proc/meminfo." In this experiment, we collect

2000 records, and each of them is a tuple of the form ($\text{Utilization}_{\text{CPU}}$ + $\text{UtilizationM}_{\text{EM}}$ + Power). Details of the records are in Figure 5. Based on the above records, we apply linear lasso regression, polynomial lasso regression, polynomial exponential lasso regression, and support vector regression, four regression models to predict the energy consumption.

4.2.3. Results. For the four regression models, we compared their relative error and the difference between predicted value and real value, and the results are in Figure 6.

From Figure 6 we can see that linear regression has very low predicting accuracy and its error is bigger than others obviously. In addition, polynomial lasso regression and polynomial exponential lasso regression models have very similar predicting results, and most of them are between −10% and 10%. In the four regression models, support vector regression has the best predicting results, and its error is between −6% and 10%.

5. Conclusion

In this paper, we study the energy consumption of data center by computing nodes. For each computing node, we analyzed both performance counter based model and system utilization based model. For performance counter based model, we find that linear method is good enough to model energy consumption; for system utilization based model, we find that nonlinear model is more accurate than linear model and support vector regression is better than polynomial regressions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported in part by the science and technology planning project of Hebei province (no. 12210334).

References

[1] A. Hooper, "Green computing," *Communication of the ACM*, vol. 51, no. 10, pp. 11–13, 2008.

[2] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proceedings of the 5th International Joint Conference on INC, IMS and IDC (NCM '09)*, pp. 44–51, IEEE, Seoul, Republic of Korea, August 2009.

[3] P. Buxmann, T. Hess, and D. W. I. S. Lehmann, "Software as a service," *Wirtschaftsinformatik*, vol. 50, no. 6, pp. 500–503, 2008.

[4] E. Keller and J. Rexford, "The 'platform as a service' model for networking," in *Proceedings of the Internet Network Management Conference on Research on Enterprise Networking (INM/WREN '10)*, p. 4, USENIX Association, San Jose, Calif, USA, April 2010.

[5] S. Bhardwaj, L. Jain, and S. Jain, "Cloud computing: a study of infrastructure as a service (IAAS)," *International Journal of Engineering and Information Technology*, vol. 2, no. 1, pp. 60–63, 2010.

[6] C. Dupont, G. Giuliani, F. Hermenier, T. Schulze, and A. Somov, "An energy aware framework for virtual machine placement in cloud federated data centres," in *Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy '12)*, pp. 1–10, IEEE, Madrid, Spain, May 2012.

[7] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 13–23, 2007.

[8] C.-H. Hsu and S. W. Poole, "Power signature analysis of the SPECpower_ssj2008 benchmark," in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS '11)*, pp. 227–236, Austin, Tex, USA, April 2011.

[9] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[10] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-System power analysis and modeling for server environments," in *Proceedings of the 14th Interntaional Symposium on Computer Architecture (ISCA '06)*, pp. 70–77, IEEE, Boston, Mass, USA, June 2006.

[11] A. W. Lewis, S. Ghosh, and N.-F. Tzeng, "Run-time energy consumption estimation based on workload in server systems," in *Proceedings of the Conference on Power Aware Computing and Systems (HotPower '08)*, pp. 17–21, December 2008.

[12] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: a packet-level simulator of energy-aware cloud computing data centers," *The Journal of Supercomputing*, vol. 62, no. 3, pp. 1263–1283, 2012.

[13] D. Kliazovich, P. Bouvry, and S. U. Khan, "DENS: data center energy-efficient network-aware scheduling," *Cluster Computing*, vol. 16, no. 1, pp. 65–75, 2013.

[14] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *The Journal of Supercomputing*, vol. 60, no. 2, pp. 268–280, 2012.

[15] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: methodology and empirical data," in *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '03)*, pp. 93–104, IEEE Computer Society, San Diego, Calif, USA, December 2003.

[16] W. Li, H. Yang, Z. Luan, and D. Qian, "Energy prediction for MapReduce workloads," in *Proceedings of the 9th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC '11)*, pp. 443–448, IEEE, Sydney, Australia, December 2011.

[17] C. Lively, X. Wu, V. Taylor 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. 245–253, 2012.

[18] C. Lively, X. Wu, V. Taylor, S. Moore, H.-C. Chang, and K. Cameron, "Energy and performance characteristics of different parallel implementations of scientific applications on multicore systems," *International Journal of High Performance Computing Applications*, vol. 25, no. 3, pp. 342–350, 2011.

[19] F. Farahnakian, T. Pahikkala, P. Liljeberg, and J. Plosila, "Energy aware consolidation algorithm based on K-nearest neighbor regression for cloud data centers," in *Proceedings of the*

6th IEEE/ACM International Conference on Utility and Cloud Computing (UCC '13), pp. 256–259, IEEE, Dresden, Germany, December 2013.

[20] J. Levon, P. Elie, and Oprofile, "A system-wide profiler for Linux systems," 2006, http://oprofile.sourceforge.net.

[21] V. M. Weaver, "Linux perf—event features and overhead," in *Proceedings of the 2nd International Workshop on Performance Analysis of Workload Optimized Systems (FastPath '13)*, p. 80, April 2013.

[22] S. K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012–1023, 2013.

[23] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing—Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2007.

Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration