Modeling the Power Consumption of Multiple Typical Applications Based on Linux

Xiaoying Wang[1,*], Kunchang Li[1], Guojing Zhang[1] and Lei Zhang[1,2]
[1]State Key Laboratory of Plateau Ecology and Agriculture, Department of Computer Technology and Applications, Qinghai University, Xining, China, 810016
[2]College of Computer Science, Sichuan University, Chengdu, China, 610064
[*]Corresponding author

***Abstract*—Since huge power consumption of large data centers has become a crucial problem recently, power models, especially precise models, turn out to be important for service providers to learn about the application status in order to make wise decisions. In this paper, we focus on the power modeling of typical applications upon Linux platforms, including CPU-intensive applications, memory-intensive applications, and network-intensive applications. We established models for these different types of applications respectively based on the collection of massive realistic data and further calibrated these models. Error analysis was also given after comparing the computed values with the actual measured data. Finally, synthetic model for the hybrid application execution scenario was figured out and discussed.**

Keywords-power model; typical applications; energy consumption modeling; Linux applications

I. INTRODUCTION

As the rapid development of Internet and information technology, the energy consumed by IT devices has become more and more unneglectable and leads to significant impact on the global carbon emission footprint. While the energy efficiency of ICT (Information Communication Technology) hardware has been dramatically improving, the overall energy used for ICT is still increasing. Worldwide per capita ICT electricity consumption exceeded 100kWh/year in 2007 and is further increasing [1]. It is estimated that the average energy consumption of ICTwill reach more than 5 times of the amount in the year 2006, and the power consumption of Internet devices will even reach to 13 times [2]. As a result, how to save energy waste and utilize the energy more effectively becomes an important and crucial issue to study about.

In order to conduct research on energy saving, we have to learn deeply about the energy consumption derivation sources. Here, in this paper, we focus on establishing the power models based on case studies of multiple typical kinds of applications upon the Linux operating system. We collect a series of data based on the real testbed, and analyze these data to set up the power model after calibration, for each type of application. Finally, the synthetic model could be established and verified. The power consumption model is necessary and useful for further study about energy-aware resource management and task scheduling.

II. RELATED WORK

The issues about energy saving and power-aware management have attracted a lot of researchers in recent years. Among these, most relevant research put insights into the power management approaches aiming at CPU, since it usually consumes most power of the entire computer system.

Frank Bellosa is the first person who tried to establish the power model based on hardware events [3].He attempted to estimated and verified the relationship between hardware events and power consumption based on experiments, and define it as linear relationship model. Then, thread-level power consumption was estimated based on hardware activities.

Power models based on performance counters could be classified into two main categories [4]: 1) Top-down counter-based modeling method, which uses only a few hardware devices to build a system-level power model; 2) Bottom-up counter-based modeling method, which collects as much as information from massive hardware event training and thus plenty of power-related factors could be gathered in order to reflect the estimated value of the applications. McCullough et al. [5] evaluated the need for pervasive power instrumentation by exploring the effectiveness of power modeling on modern hardware.Joseph et al.[6] examined the use of hardware performance counters as proxies for power meters and discussed which performance counters count power-relevant events. Lefurgy et al. [7] presented a technique that could control the peak power consumption of a high-density server by implementing a feedback controller that uses system-level power measurement to select the highest performance state while keeping the system within a fixed power constraint.

Different from the above prior research, in this paper, we focus on the power modeling for typical Linux applications. We chose three different types of applications and set up their models respectively, in order to estimate the power consumption values. Different characteristics of the application types were incorporated into consideration.

III. THREE TYPICAL APPLICATIONS

In this paper, we selected three typical applications and run them on Linux testbed to gather necessary data. This section introduces these three applications.

A. Typical CPU-Intensive Application

Here, we chose *ubench CPU benchmark* as a typical application of the CPU-intensive type. *Ubench* is the Unix Benchmark Utility which attempts to provide a single measure of performance for machines running various flavors of Unix operating system [8]. *Ubench* will execute rather senseless mathematical integer and floating-point calculations for 3 minutes concurrently using several processes, and the ratio of floating-point calculations to integer is about 1:3. Specifically, *ubench* will spawn about 2 concurrent processes for each CPU available on the system. This ensures all available raw CPU horsepower is used.

According to the above features of *ubench*, we use it as the typical CPU-intensive application for later experiments.

B. Typical I/O-Intensive Application

Here, we chose *bonnie++* benchmark as a typical application of the I/O-intensive type. *Bonnie++* is a free file system benchmarking tool for Unix-likeoperating systems, aimed at performing a number of simple tests of hard drive and file system performance [9]. The main program tests database type access to a single file, and it tests creation, reading, and deleting of small files which can simulate the usage of programs such as Squid, INN, or Maildir format email, and then a brief report will be produced. It requires that the file for test should better be twice the size of available RAM.

According to the above features of *bonnie++*, we use it as the typical I/O-intensive application for later experiments.

C. Typical Network-Intensive Application

Here, we chose the *netperf* benchmark as a typical application of the network-intensive type. *Netperf* is a software application that provides network bandwidth testing between two hosts on a network [10]. It supports Unix domain sockets, TCP, SCTP, DLPI and UDP via BSD Sockets. *Netperf* provides a number of predefined tests, for example, to measure bulk (unidirectional) data transfer or request response performance. It requires client/server mode and will start up a virtual server during the tests.

According to the above features of *netperf*, we use it as the typical network-intensive application for later experiments. We choose five performance metrics from the network tests, including TCP_STREAM, UDP_STREAM, TCP_RR, TCP_CRR and UDP_RR.

IV. DATA COLLECTION AND ANALYSIS

In order to collect necessary realistic data, we have built a testbed incorporating data collection devices and also servers running several applications. From the testbed, we can get one data point every 10 minutes, and the monitoring tool will draw a curve using its GUI (Graphical User Interface), one example as shown in Figure I. Hereafter, we user 24 hours as the time period length, unless other value specified. Besides, we use one DELL R720 server as the host to run the applications, which has CentOS preinstalled.

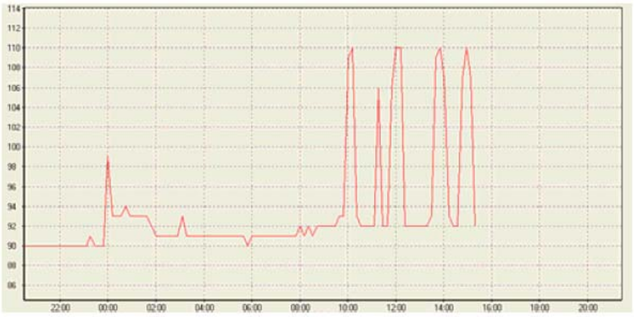


FIGURE I. SNAPSHOT OF THE MONITORING TOOL

A. Measuring Data under the Idle State

As we know, the server will consume some power even if there is no task running on it, since it needs electricity to keep alive and also keep the operating system running. Here, we first collect the data under idle state to learn about the basic power consumption without any additional target application executed.

Figure II shows the data we collected on Apr. 21st, 2016, from which we can see that the power consumption in idle state keeps constant around 90W. Based on more data throughout a whole month, we take denote p_0=90Was the basic power consumption value for the target server system.

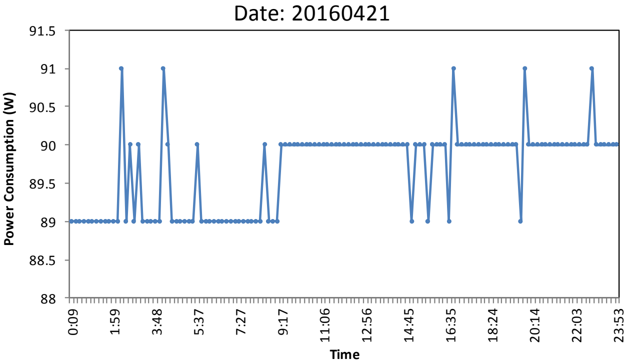


FIGURE II. POWER CONSUMPTION IN IDLE STATE

B. Collecting Data for CPU-Intensive Applications

The following experiment time period is set up 24 hours. First, we run a CPU-intensive task from 9:00 to 21:00 repeatedly. The power consumption measured during this period is shown in Figure III, from which we can see that the execution of the task has remarkable impact on the consumed power.

C. Collecting Data for I/O-Intensive Applications

Then, we run an I/O-intensive task from 9:00 to 21:00 repeatedly in a whole day. The power consumption measured during this period is shown in Figure IV, from which we can see that the execution of the task also has some impact on the consumed power, but not as large as CPU-intensive tasks.

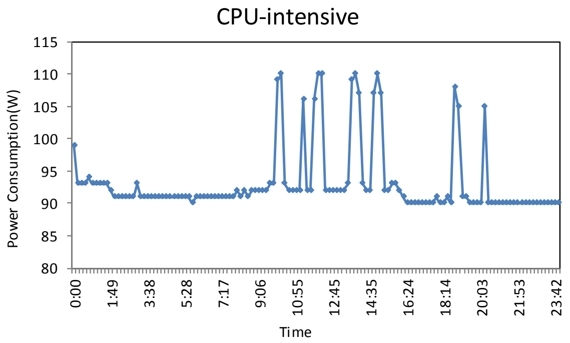


FIGURE III. POWER CONSUMPTION VARIATION OF CPU-INTENSIVE APPLICATIONS

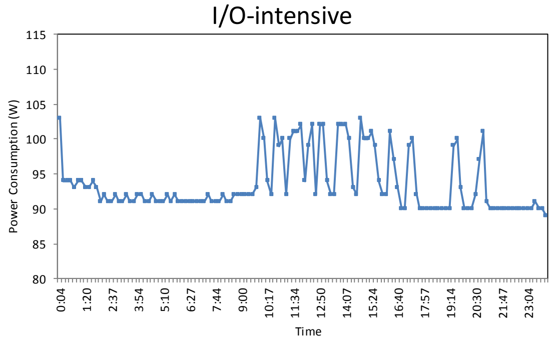


FIGURE IV. POWER CONSUMPTION VARIATION OF I/O-INTENSIVE APPLICATIONS

D. Collecting Data for Network-Intensive Applications

Furthermore, after we run a network-intensive task from 9:00 to 21:00 repeatedly, the power consumption measured during that day is shown in Figure V. The variation amount of power consumption is similar with the I/O intensive scenario. We can see that the power consumed rose up above 100W as the task started and went back to p_0 as the task ended.

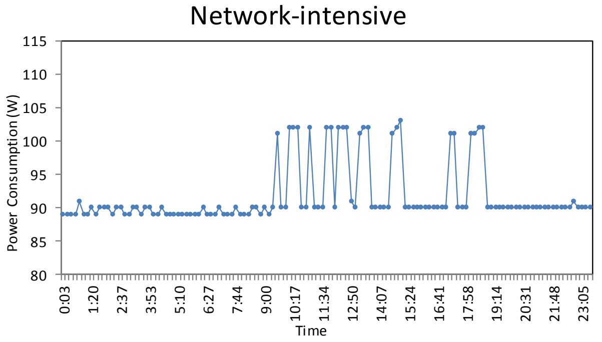


FIGURE V. POWER CONSUMPTION VARIATION OF NETWORK-INTENSIVE APPLICATIONS

V. Model Establishment and Validation

In this section, we present the procedure of establishing power models, including data input, power estimation, and error analysis.

A. Considering Only CPU Power Consumption

An accurate model for estimating CPU power consumption has to incorporate all kinds of detailed factors, but getting these details will definitely affect the normal work of the original system. Thus, Kansal et al. [11] proposed a light-weight alternative method to trace the CPU utilization and calculate the consumed power in a linear way, as follows:

$$E_{cpu}=\int_{t1}^{t2} P_{cpu}\,dt \tag{1}$$

$$Pcpu = \alpha cpu*ucpu +p0 \tag{2}$$

where $\alpha_{cpu}$is a certain coefficient, which can be calculated by fitting.

B. Multivariable Synthetic Model

For I/O-intensive and network-intensive applications, only considering CPU power is not enough to set up an appropriate model. We have to integrate the disk load and network load into the model as well, as follows.

$$Edisk(T) = \int_{t1}^{t2} P_{IO}\,dt \tag{3}$$

$$PIO\ = \alpha *\ ucpu +\beta *\ umem +\gamma *bwr +p0 \tag{4}$$

where P_{IO} denotes the average power consumed, u_{mem} is the mean utilization of the physical memory, and b_{wr} denotes the total bytes of read and write operations. Similarly, α, β and γ are coefficient in the model and could be calculated.

In order to collect realistic data for curve fitting, we run *bonnie++* and *netperf* on the target node. During the experiments, bandwidth data of both I/O and network were recorded. By monitoring and measuring the above metrics, where Table I shows some samples of the measured data, the value of the three coefficients could be computed out, and thus the model obtained is as follows:

$$Edisk(T)=\int_{t1}^{t2}\left(147.055\cdot u_{cpu} + (-24.467)\cdot u_{mem} + 0.000022\cdot b_{wr} + 90\right)dt \tag{5}$$

TABLE I. SAMPLE OF MEASURED DATA

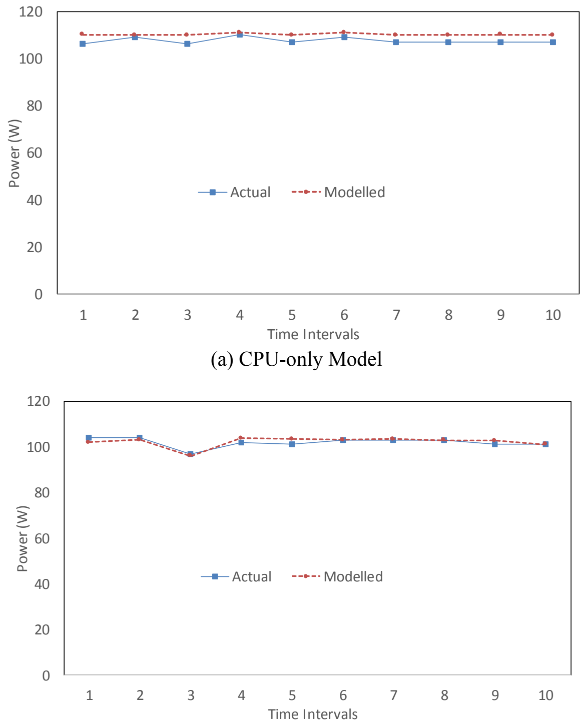
	Group 1	**Group 2**	**Group 3**	**Group 4**
u_{cpu}	22.17%	22.86%	23.31%	23.50%
u_{mem}	97.87%	97.93%	97.95%	98.08%
b_{wr}	15640MB	15640MB	31280MB	31280MB

C. Model Validation and Error Analysis

After establishing the models, we have to verify the model to see whether they are applicable to use in realistic environments. Here, we executed multiple applications simultaneously upon the testbed in two days, divided into ten time intervals, and recorded the actual power consumption values during the whole procedure. Figure VI show results from the 10 time intervals during the experiment period.

First, we use the CPU-only model to estimate the power consumption values and compare them to the actual measured values. The result is shown in Figure VI(a). The mean error of this model results in 2.45% here.

Furthermore, the syntheticmodel is adopted to estimate the power consumption values. The result is shown in Figure VI(b). As we can see, the calculated value is very close to the actual measured ones, with the mean error of 0.197%, which illustrates the feasibility and the accuracy of the model.



(a) CPU-only Model

(b) Synthetic Model

FIGURE VI. COMPARISON OF MODELLED AND MEASURED VALUES

From another point of view, although the CPU-only model is less accurate compared to the other one, the resulted error is still relatively small in an acceptable level. This implies that CPU is still the dominant power consumer inside the computer system. The advantage of the CPU-only model is that it's very simple and easy to establish and calibrate.

VI. Conclusion and Future Work

In this paper, we focus on the problem of how to establish power consumption models for sever nodes, which is a very important issue for sustainable datacenter management and maintenance, as the huge energy footprint brought by large-scale datacenter in recent years. Three main contents we presented include: 1) selecting typical applications and collecting data; 2) setting up the formal model and 3) data fitting and analysis. First we figure out the CPU-only model as a baseline, which is simple and easy to set up. Then, the synthetic model can be obtained by fitting more kinds of data into the equations. Model validation experiments show that the CPU-only model can estimate the power consumption value in an error range of 2.45%, while the synthetic model behaves better and results in an error range of 0.198%.

As a part of ongoing work, we attempt to test more kinds of applications to verify and calibrate the models, and try to use these models for designing efficient energy-aware management strategies for datacenters in the future.

Acknowledgment

This paper is partially supported by The National Natural Science Foundation of China (No. 61363019, No. 61440021, No. 61272087, No. 61073008 and No.61563044); National Natural Science Foundation of Qinghai Province (No. 2014-ZJ-718, No. 2015-ZJ-725).

References

[1] B. Aebischer and L. M. Hilty, "The Energy Demand of ICT: A Historical Perspective and Current Methodological Challenges," in ICT Innovations for Sustainability, M. L. Hilty and B. Aebischer, Eds. Cham: Springer International Publishing, 2015, pp. 71–103.

[2] P. Schulte, H. Welsch, and S. Rexhäuser, "ICT and the Demand for Energy: Evidence from OECD Countries," Environ. Resour. Econ., vol. 63, no. 1, pp. 119–146, 2016.

[3] S. Kellner and F. Bellosa, "Energy Accounting Support in TinyOS," PIK - Prax. der Informationsverarbeitung und Kommun., vol. 32, no. 2, 2009.

[4] R. Bertran, M.Alez, X. Martorell, N. Navarro, and E. Ayguade, "Counter-based power modeling methods: Top-down vs. bottom-Up," Comput. J., vol. 56, no. 2, pp. 198–213, 2013.

[5] J. C. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, and R. K. Gupta, "Evaluating the effectiveness of model-based power characterization," Proc. 2011 USENIX Conf. USENIX Annu. Tech. Conf., p. 12, 2011.

[6] R. Joseph and M. Martonosi, "Run-time power estimation in high performance microprocessors," in Proceedings of the 2001 international symposium on Low power electronics and design - ISLPED '01, 2001, pp. 135–140.

[7] C. Lefurgy, X. Wang, and M. Ware, "Server-level power control," in Fourth International Conference on Autonomic Computing, ICAC'07, 2007.

[8] Ubench homepage. http://www.phystech.com/download/ubench.html

[9] Bonnie++ benchmark homepage. http://www.coker.com.au/bonnie++/

[10] Netperf homepage: http://www.netperf.org/netperf/

[11] Kansal A, Zhao F, Liu J, Kothari N, Bhattacharya A. Virtual machine power metering andprovisioning.Proceedings of the listACM Symposium on Cloud Computing. Indi-anapolis, USA, 2010:39-50.