Accurate Multicore Processor Power Models for Power-Aware Resource Management

Ibrahim Takouna, Wesam Dawoud, and Christoph Meinel
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
{*ibrahim.takouna, wesam.dawoud, christoph.meinel*}@*hpi.uni-potsdam.de*

***Abstract*—Power management is one of the biggest challenges facing current datacenters. As processors consume the dominant amount of power in computer systems, power management of multicore processors is extremely significant. An efficient power model that accurately predict the power consumption of a processor is required to develop effective power management techniques. However, this challenge rises with using virtualization and increasing number of cores in the processors.**

In this paper, we analyze power consumption of a multicore processor; we develop three statistical CPU-Power models based on the number of active cores and average running frequency using a multiple liner regression. Our models are built upon a virtualized server. The models are validated statistically and experimentally. Statistically, our models cover 97% of system variations. Furthermore, we test our models with different workloads and three benchmarks. The results show that our models achieve better performance compared to the recently proposed model for power management in virtualized environments. Our models provide highly accurate predictions for un-sampled combinations of frequency and cores; 95% of the predicted values have less than 7% error. Thus, we can integrate these models into power management mechanisms for a dynamic configuration of a virtual machine in terms of the number of its virtual-CPUs and the frequency of physical cores to achieve both performance and power constrains.

***Keywords*-power; management; virtualization; modeling; multicore;**

I. Introduction

Datacenter power consumption has become a significant concern with the rapid emergence of cloud services such as Amazon EC2. For example, Hamilton [1] has reported that Amazon's datacenters are facing a highly increased power demand where the servers consume 59% of the total power supply. Furthermore, the U.S. Environmental Protection Agency (EPA) reported that the energy consumption of the datacenters located on U.S. consumed 61 billion kilowatt-hours in 2006 which costs $4.5 billion [2]. Thus, there have been many proposed approaches for datacenters power management [3][4]. Current datacenters consist of a number of servers leveraging multicore processors. According to Moore's law, industry could double the number of cores in a single processor every 18 months. Unfortunately, They also doubled the power density of the processor. The processor is the component that consumes the most dynamic power of a computer system [5][2]. Nevertheless, an idle sever consumes over 50% of its peak power [6] which means that a server with low utilization is very power-inefficient. Hence, virtualization technology has been rapidly employed in datacenters to increase servers' utilization by enabling applications consolidation onto a fewer number of physical servers and turning off unused servers to save power. There are several proposed approaches for power management. Mostly, these approaches consider the CPU frequency and CPU utilization to build power models. For instance, Urgaonkar et al. [7] and Gandhi et al. [8] have adopted non-liner quadric models of power consumption for power management in virtualized environments. The relationship between power consumption of multicore processor and frequency cannot be covered with one fitting curve, as we will see in Section II. Importantly, setting frequency of a multicore processor as a unit without considering the real active frequency and number of cores will lead to serious errors.

Moreover, Fan et al. [10] have included CPU utilization in their proposed power model. However, using utilization to build a power model for a multicore processor could be inaccurate, because the power consumed by a multicore processor with one active core with 100% utilization is more than the power consumed by two active cores each of them 50% utilized for the same workload. We found this result by conducting an experiment using a virtual machine (VM) with a multithreaded application. This VM ran with 1 virtual CPU and had 100% utilization and only ran on one physical core. In this scenario, the power consumed by the physical CPU was 26 watts. On the other hand, when the VM ran with 2 virtual CPUs and had the same total CPU utilization 100%, in this scenario, the physical CPU just consumed 17 watts, and each core was 50% utilized. Importantly, both of the configurations gave the same performance. Indeed, the latter could be better due to exploiting the multithreading. Thus, we conclude that using only CPU frequency or CPU

978-0-7695-4612-4/11 $26.00 © 2011 IEEE
DOI 10.1109/DASC.2011.85

utilization as an input for power modeling can be inefficient, in particular, for power estimation of multicore processors.

Evolving virtualized environments enables consolidation of multithreaded High Performance Computing (HPC) applications; these applications can efficiently utilize multicore processors. However, to implement power-aware resource management techniques for such environments an accurate power estimation model is required. Hence, the purpose of this work is to build CPU-Power consumption models that accurately estimate the power consumption of virtualized servers with multicore processor. These models can be used by power-aware resource management techniques to achieve better power savings. Our work is distinct from others as follows. This paper presents CPU-Power consumption models taking into account number of the actual active cores N and average running clock frequency F at each sample. It analyzes and evaluates the performance of our proposed models statistically and experimentally. The statistical analysis using the regression R^2 indicates that our models can cover more than 97% of system variations. Experimentally, our proposed models achieve better performance compared to the model adopted by [7][8]. We evaluate models using three different applications with different characteristics (i.e., CPU-intensive, Memory-intensive, and IO-intensive). The results show that 95% of the predicted values have less than 7% error. Furthermore, the maximum prediction error is less than 4% error for Memory-intensive and IO-intensive applications. As future work, we will use these models to build a dynamic optimizer that optimizes the number of cores and their frequency settings and dynamically configures a VM to cope with a workload and meet power consumption constrain

The rest of this paper is organized as follows. The following section presents a development of CPU-power model. Section III presents a statistical analysis evaluation of the power models. Section IV shows experimental performance and results. The related work is presented in Section V. Finally, our conclusions and future work are presented in Section VI.

II. CPU-Power Model development

Several works have used linear models to represent the power consumption of a system or just a processor. These models are based on CPU utilization or other concerned resources such as memory. Current processors have multiple cores, which can operate at different frequency levels at runtime using DVFS. Hence, in this section we discuss the relationship between the CPU-Power consumption and CPU-frequency from one side, and the CPU-Power consumption and number of active cores from the other side. However, first we present experimental setup and measurement tools.

A. Experimental setup and measurement

The evaluation experiments were performed on Fujitsu PRIMERGY RX300 S5 server, which has a CPU-Power measurement capability. It has a processor of Intel(R) Xeon(R) CPU E5540 with 4-cores. The frequency ranges from 1.59GHz to 2.53GHz. Each core enables 2-logical cores. The server is equipped with 12GB physical memory. The experiments were run on a virtualized server using Xen-4.1 hypervisor.

To build our models, we used a CPU-intensive benchmark EP Embarrassing Parallel, which is one of NAS Parallel Benchmarks (NPB) [11]. It is a multithreaded benchmark, which runs a number of threads corresponding to the number of virtual CPU of a virtual machine. To evaluate our models, we used CG and BT benchmarks of NPB suite. BT benchmark and CG benchmark are IO-intensive and Memory-intensive, respectively. More detail about the characteristics of NPB benchmarks is found in [12].

Xenpm tool [13] was used to measure average running frequency and number of active cores. Fig. 1 summarizes the system overview. Importantly, Fig. 1 depicts an output of xenpm showing two cores running on two different frequencies. Furthermore, it illustrates the change of average frequency, performance states (P0-P8), and sleeping states (C0-C3). Thus, using xenpm to measure the real active frequency provides more accuracy to our models. Finally, we used the CPU-Power measurement capability of our server to measure the power consumption of the CPU. In our experiments, the percentile average was considered to get accurate power readings.

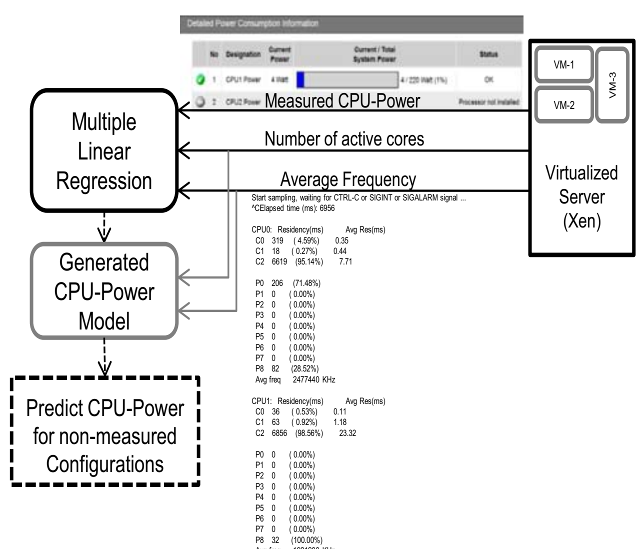


Figure 1: Overview of the system.

B. CPU-Power and frequency relationship

CPU-Power consumption is composed of dynamic and static power. The dynamic power is proportional to the cube of frequency F [14]. The dynamic power is the important factor for reducing power consumption using DVFS technique, which results in slowdown the performance. To obtain the relationship between CPU-Power consumption and frequency, we ran EP-NPB CPU-intensive benchmark on a virtual machine at different CPU frequencies. We measured the power consumption only for the CPU. Thus, we obtained the curves in Fig. 2. Then, by applying a linear regression model, we found that the best relationship could be fitted in polynomial linear with regression $R_2 = 0.99$. We generalized it as a quadric model in (1) which resembles the proposed model by [7][8] to estimate the power consumption of a server. θ and P_{min} are constants that should be chosen to achieve an accurate prediction.

$$P_{(F)} = P_{min} + \theta(F - F_{min})^2 \tag{1}$$

Although Fig. 2 shows perfect fitting for each curve of a number of cores, the total system variations cannot be covered with considering only the frequency. Hence, we need different values of θ and P_{min} at each number of active cores. For example, to estimate CPU-Power at frequency 1.72 GHz when 8 active cores using (1) the best values for θ and P_{min} are 37 watts and 37 $Watt/GHz^2$ respectively. The estimated power is 37.6. It is approximately equal to the measured value 38 watts. Nevertheless, The values of θ and P_{min} should be adapted again to predict the power when just 4 cores are active. Accordingly, we study the relationship between the power consumption and number of active cores in next section.

C. CPU-Power and number of active cores relationship

To estimate the power consumption of multicore processors, we found that it is important to study the relationship between CPU-Power and number of active cores. To this end, we obtained the curves in Fig. 3. The curves have a linear trend line. The relationship is well approximated by a linear model with regression $R^2 = 0.95$, which means that the power consumption and number of active cores have a strong linear association and can be represented by (2). N is the number of active cores, and P_{min} is the power consumed by one core running at frequency F. α is the slope of the power-to-active cores curve at frequency F. Importantly, each curve has two different slopes. The first one is when the number of active cores is less than 4 cores; the other one is when the number of active cores is more than 4 cores. Moreover, the first slope is greater than the second one. The main reason of this case was that we had a processor with 4 physical cores. Each physical core has two logic cores, and the power consumed by a logical core is less than the power consumed by a physical core.

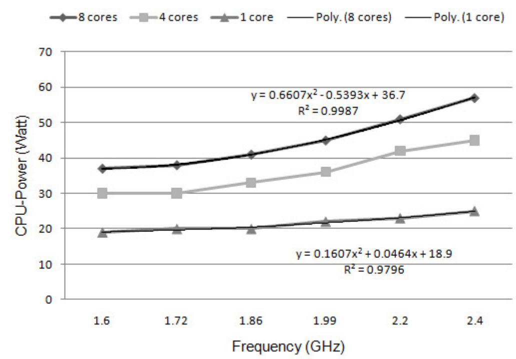


Figure 2: CPU-Power consumption relationship with frequency.

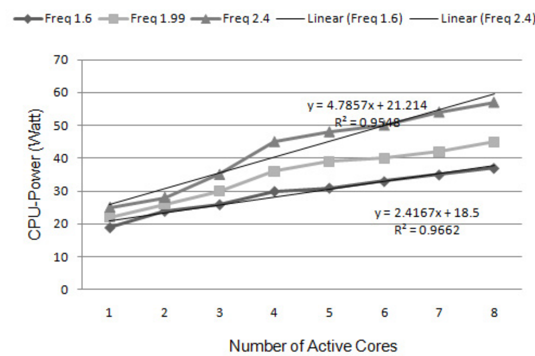


Figure 3: CPU-Power consumption relationship with number of active cores.

$$P_{(N)} = P_{min} + \alpha.N \tag{2}$$

D. CPU-Power estimation models

From previous sections, we found a strong relationship between CPU-Power consumption and both frequency and number of active cores. In this section, we refer to the model adopted by [7][8] as Model-0. Model-0 is represented by (3); it does not include the number of active cores. Our first model is denoted by Model-1. Model-1 presented in (4) is a multiple linear regression with the intercept constant C. Furthermore, Model-0 consists of two components: the frequency and the number of cores. These two components mainly contribute to multicore processor's power. Equation 5 represents Model-2. Model-2 is similar to Model-1, but its intercept constant C is zero. Finally, we removed the first degree term of frequency of Model-2; we obtained Model-3 represented by (6). However, we will study the predication accuracy of these models showing the worst and the best cases for each model. These models were built using training samples of the frequency and the number of cores combinations; some other combinations (i.e., un-sampled

Table I: The determined values of CPU-Power models coefficients and statistics.

Model	θ_2-range	θ_2	θ_1-range	θ_1	α-range	α	C-range	C	Std. Err.	R^2
0. (3)	-29.98 - 42.6	6.31	-152.8 - 138.25	-7.27	0	0	-118.86 - 168.24	24.68	7.95	0.291
1. (4)	-0.14 - 12.77	6.31	-33.17 - 18.63	-7.27	3.12 - 3.48	3.3	-15.77 - 35.36	9.79	1.41	0.978
2. (5)	3.08 - 4.63	3.8	0.88 - 4.37	2.63	3.13 - 3.49	3.31	0	0	1.40	0.998
3. (6)	4.78 - 5.20	4.99	0	0	3.26 - 3.60	3.43	0	0	1.53	0.998

data) were used to validate our models. Furthermore, we used only a CPU-intensive EP benchmark in training stage.

$$P_{(F,N)} = \theta_2.F^2 + \theta_1.F + C \tag{3}$$

$$P_{(F,N)} = \theta_2.F^2 + \theta_1.F + \alpha.N + C \tag{4}$$

$$P_{(F,N)} = \theta_2.F^2 + \theta_1.F + \alpha.N \tag{5}$$

$$P_{(F,N)} = \theta_2.F^2 + \alpha.N \tag{6}$$

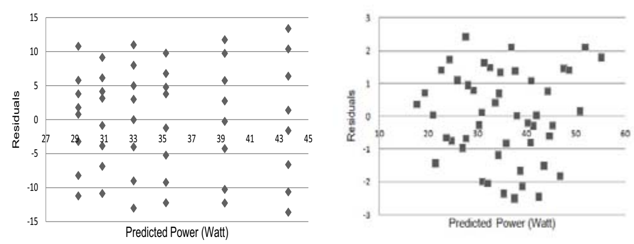
III. Statistical analysis

This section discusses some statistical analysis of our CPU-Power estimation models focusing on Model-1 to show its efficiency to predict the power consumption of a processor. We used plots to check models linearity and normality assumptions [9].

First, we tested the models linearity using a plot of residuals versus predicted values. Fig.4-(a) represents a residuals plot of Model-0. This plot shows a certain pattern indicating that the model makes systematic errors whenever the number of cores varies from 1 to 8. In Fig.4-(a), if we consider the first left vertical residuals points which represent residuals of frequency 1.6GHz, we find that the residuals values increase negatively when few of cores are active (e.g., the residual value is -11 when one core is active). On the other hand, the residuals values increase positively when the number of active cores is more than 4 cores. The reason was the over-estimation of CPU-Power with enabled logical cores. Moreover, the predicted power is limited by frequency. Thus, it gives the short range [29, 43]. Consequently, it might yield to significant errors when used for predicting un-sampled data that were not used in training stage. Furthermore, the residual rang [-15, 15] of Model-0 is wider than the residual Model-1's range [-3, 3], which means that our models are more accurate compared to Model-0. Additionally, our models are not limited by frequency that make them scalable with the number of cores in the processor. Fig.4-(b) is residuals of Model-1. The points are symmetrically distributed around a horizontal line. This proves that our model satisfies the linearity assumption of the liner regression [9]. From this test, we conclude that our models can predict beyond the range of the sample data without errors.

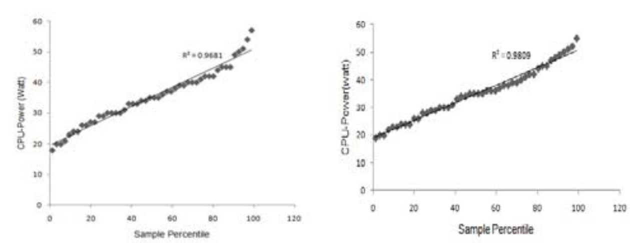
Second, we tested the models against normality using a normal probability plot of the residuals. Fig. 5-a and 5-b show similar plots of predicted and sample percentile points. These points are very close to the diagonal line which means that the residuals of models are normally distributed.

Table I summarizes the determined values of CPU-Power models coefficients and statistics. This table shows the ranges of the models coefficient. Model-2 shows a small range for its coefficients. To illustrate, θ_2-range (i.e., [2.1, 5.10]) is very small compared to Model-0's θ_2-range (i.e., [-33.54, 42.9]). However, Model-2 and Model-3 are regressions with zero constant. Furthermore, the regression statistics in Table I show that the regression R^2 of our models is higher than R^2 of the Model-0. For instance, Model-2 and Model-3 have regression R^2=0.99 which means that these two models can explain 99% of the power variations. The power variations were determined by variations in the independent variables (i.e., frequency and active cores). In contrast, Model-0, which only considers frequency, has regression R^2=0.259. Model-0 explained only 25% of power variations using frequency.



(a) Model-0 (b) Model-1

Figure 4: Predicted power and residual plot.



(a) Model-0 (b) Model-1

Figure 5: Normal probability plot.

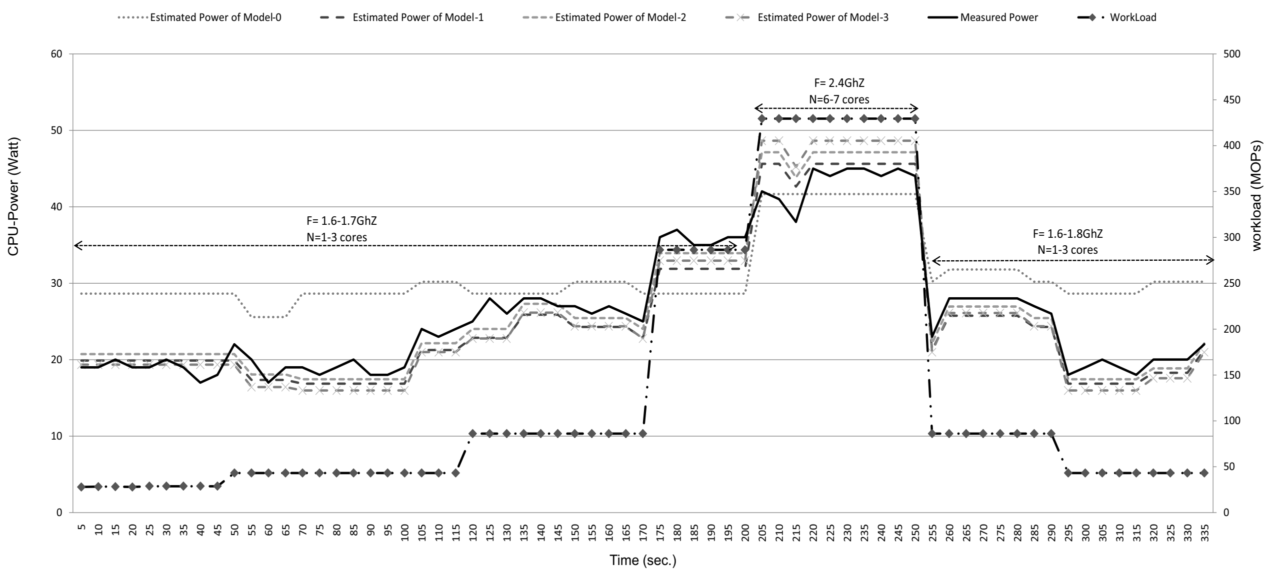


Figure 6: Trace of measured consumed CPU-Power and predicted CPU-Power for the four models.

IV. Performance evaluation

As we discussed models performance statistically earlier, in this section, we show and compare the performance of the models experimentally. To achieve this, we conducted three experiments using three different benchmarks of NPB benchmark namely: EP, CG, and BT. These benchmarks represent CPU-Intensive, Memory-Intensive, and IO-Intensive applications respectively. CG and BT benchmarks were not used in training stage.

A. CPU-intensive applications

To evaluate our models against CPU-Intensive applications, we used EP benchmark to generate a workload which was changed with time. As shown in Fig. 6, we started with a low workload which increased with time until it reached its maximum approximately at time 205 sec. Then, it started to decrease after time 250 sec. During the experiment, we measured the CPU-Power consumption every 5 seconds. Then, we computed the estimated power using the four different models. Obviously, the curve shows that Model-0 has a big difference between its estimation and the measured power when low-workload and few of cores are actives (i.e., 1-3 active cores). However, it shows a good performance in high-workload when all the cores are active. This case is similar to estimation power consumption of a processor as a unit regardless of the active cores number.

As our models include the number of active cores and the average frequency, they accurately estimated the power in both areas of workload (i.e., low-workload and high-workload). Furthermore, although Model-2 and Model-3 statistically (i.e., regression R^2) are better than Model-1, the experiment demonstrated that Model-1 with constant C achieved better performance than the other models. Finally, slight percentage of error could be observed in our models due to considering a logical core as a physical core. However, as we illustrated in section II-B, the power consumed by a logical core is less than the power consumed by a physical core.

Now, we discuss the prediction accuracy by computing the percentage of error using the following formula.

$$PoE = |(Estimiated - Measured)/Measured| * 100\%$$

Furthermore, we obtained the Empirical Cumulative Distribution Function of Percentage of Error CDF(PoE). Fig.7 depicts a plot of CDF(PoE) for each model. The x-axis represents the Percentage of Error (PoE), and y-axis shows the percentage of data points (i.e., predicted power values) that achieve error less than each value of x. For instance, 90% of the predicted values using Model-0 has less than 40%, and this error might increase to 50%. On the other hand, our models show that 90% of values were predicted with less than 9% error. Although R^2 value of Model-1 is less than R^2 value of Model-2 and Model-3, Model-1 demonstrated the best results where 95% of the predicted values had an error less than 7%. Generally, the prediction accuracy of Model-1 empirically outperforms the prediction accuracy of Model-2 and Model-3.

Significantly, our proposed models achieve high prediction accuracy due to considering both the number of active cores and the average running frequency. Additionally, the precise readings of CPU-Power that were realized using CPU-Power measurement capability of our server assisted us to build these accurate models. To test our models' ability to predict those un-sampled combinations of frequency and

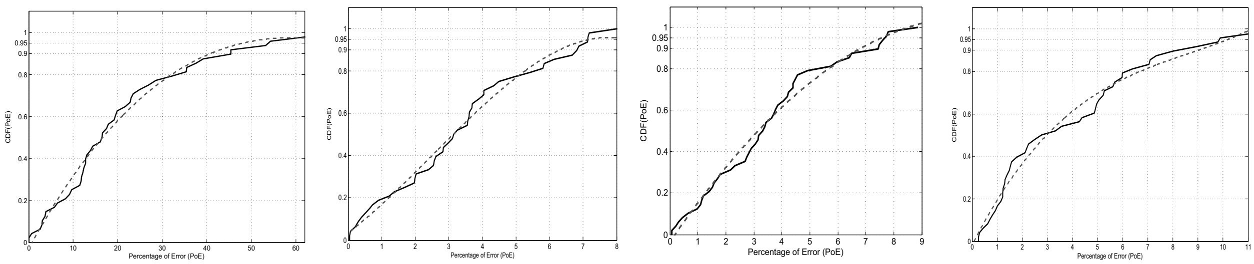


Figure 7: Prediction accuracy of the CPU-Power models.

number of cores, we conducted some experiments with different un-sampled combinations frequency 2.53GHz and number of cores. Table II presents the results of these experiments. The results proved that our models are able to predict with high accuracy even un-sampled combinations of frequency and number of cores.

B. *Memory-intensive applications*

In this section, we evaluated performance of the models for applications that are considered memory-intensive using CG benchmark. Fig. 8 shows the estimated power versus the measured power. The diagonal line represents the perfect predication line which illustrates the deviation of the estimated values from the measured values. In other words, the predicted value is equal or close to the measured value if it is one of the perfect predication line points or close to this line. From Fig. 8, the data points that represent our models lie very close to the perfect predication line. Furthermore, we computed the maximum prediction error of each Model. We found that Model-1 and Model-2 had less maximum prediction error compared to the other two models. However, with less than 6% maximum prediction error for Model-1 and Model-2, these two models are still accurate. The maximum prediction error of Model-0 was 14.4%.

C. *IO-intensive applications*

As we presented the performance of our models for CPU-intensive and Memory-intensive applications in the previous sections, this section presents performance of the models for IO-intensive applications. We repeated the experiments procedure of the previous section using BT benchmark. Fig. 9 also shows the estimated power versus the measured power. We can see that the predicted values of Model-1 are on perfect predication line or very close to it. In contrast to Model-1, Model-0 and Model-3 show a large deviation from the perfect line. Moreover, we found that Model-1 and Model-2 had less maximum prediction error compared to the other two models. Model-1 and Model-2 achieved less than 5% maximum prediction error. Finally, the maximum prediction error of Model-0 became worse with 22.07% error.

Table II: The predicted CPU-Power for un-sampled combination of frequency 2.53 GHz and number of cores.

Cores	Measured	Model-0	Model-1	Model-2	Model-3
3	42	47.09	42.27	41.23	42.06
4	45	47.09	45.57	44.53	45.46
8	58	47.09	58.77	57.73	59.06

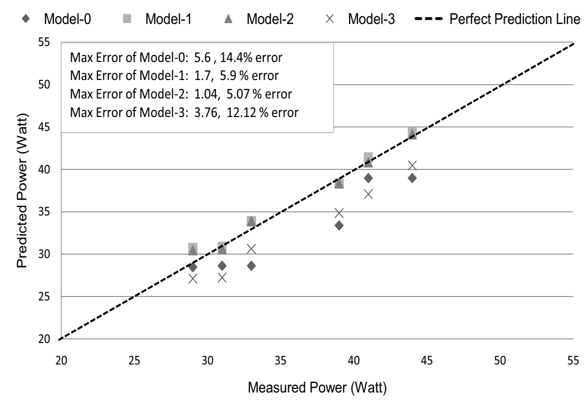


Figure 8: CPU-Power models fit for CG-Memory-Intensive: measured vs. estimated.

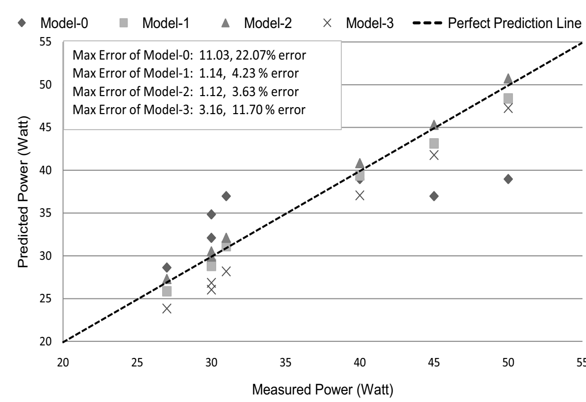


Figure 9: CPU-Power models fit for BT-IO-Intensive: measured vs. estimated.

V. Related Work

Power models could be built for a full server or for a particular component of the server such as a CPU. At the server level, Dynamic Voltage/Frequency Scaling (DVFS) mechanism is exploited to obtain power consumption proportional to the total workloads demand of VMs hosted on servers. DVSF mechanism enables processors to run on different performance states. Many papers have explored policies for reducing CPU power consumption using DVFS [15]-[18], but these approaches are not designed for virtualized servers that host multiple applications on the same physical server and allow dynamic change of virtual machine configuration.

There are several solutions implemented for virtualized environments [20]-[22]. Kusic et al. [20] have developed a dynamic resource provisioning framework based on lookahead control, which estimates the future workload demand. However, they concluded that the intensity of the workload directed at the VMs does not affect the power consumption. Additionally, they reported that the power consumed by a host machine only affected by the number of VMs running on it regardless of the arrival rate experienced by the VMs. In this paper, we show different results where the power consumption changes with the workload intensity even when we have one VM.

To implement a power management mechanism for a system, it needs to construct a power consumption model for the system. The BlackBox approach is a common approach to construct a power model for either a full-system or a single processor. Using BlackBox approach could sacrifice some accuracy, but it realizes simplicity by avoiding reliance on detailed knowledge of the hardware's implementation. We summarize some of the previous works that adopt this approach to build a power model for either a full-server or a single component such as a processor. Importantly, we used BlackBox approach to build our models. This approach totally depends on the training data. In this work, we measured both of frequency and CPU-power using precise tools.

There are several works considering power modeling of a full-server. Ranganathan et al. [24] has proposed a dynamic power budgeting optimizations. In their work, a lookup table has been built for relating power and performance to system resource utilization. Similarly, Fan et al. [10] have developed a power optimization model using a linear model based on CPU utilization and a measured power. Furthermore, a linear model has been used to facilitate server consolidation [26]. This model used CPU, memory, and disk utilization for a server. In [27], authors have constructed a model based on utilization and CPU performance counters to model the power consumption of two different servers [27]. Finally, [7], [8] have proposed a non-linear relationship between power and frequency to estimate power consumption of servers. Generally, a full-server power consumption model should be re-calibrated when applied to a different server or a different application.

On the other hand, different power models have been constructed to model the power consumption of a processor [23][28][29]. For instance, Bellosa et al. [28] and Li et al. [23] used performance counters to generate power models. They have determined a linear relationship between processor power consumption and several performance counters. A linear model of the power consumption of an Intel XScale processor and its memory system has been built in [29]. In [30], authors have built a model for power consumption at a task level based on CPU cycles and memory cycles executed for different types of tasks in operating system; however, they did not consider a multithreaded tasks and multicore processors. Finally, Ben-Itzhak et al. [5] has constructed a simple linear power model for multithreaded applications. Their model has been used to achieve power-aware thread allocation solutions. In contrast, we take into account the ability of a core to be run with different frequency levels.

VI. Conclusions and Future work

In this paper, we have developed models to estimate the power consumption of multicore processors. Our work is distinguished from previous work by considering both the number of active cores and the frequency of multicore processor. We developed our prediction models using an Intel(R) Xeon(R) CPU E5540 processor. We validated our proposed models statistically and experimentally with varied workloads. The results of the experiment showed that our model achieved high accuracy of CPU-Power estimation. Furthermore, we applied our model to predict power consumption of different applications that are characterized as CPU, Memory, and IO intensive. Then, we compared the worst predication error of these models. The results indicated that our models in particularly Model-1 and Mode-2 had a stronger robustness than Model-0.

Hence, our proposed models can be applied to dynamic power-aware configuration of cores' frequency and virtual machines' vCPU number. This enables new adaptive power management solutions for virtualized servers. Furthermore, they can be used to realize a fine-grained power provisioning proportional to workloads.

Acknowledgments

This work has been done by utilizing resources of HPI Future SOC-Lab that its industrial partners are Fujitsu and Hewlett Packard.

References

[1] J. Hamilton, "Cooperative expendable micro-slice servers (CEMS): low cost, low power servers for internet-scale services," *In conference on Innovative Data Systems Research CIDR'09*, January 2009.

[2] U.S. Environmental Protection Agency. Epa report on server and data center energy efficiency. ENERGY STAR Program, 2007.

[3] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No power struggles: coordinated multi-level power management for the data center," *In Proceedings of ASPLOS*, March 2008.

[4] R. Urgaonkar, U.C. Kozat, K. Igarashi, and M.J. Neely, "Dynamic resource allocation and power management in virtualized data centers," *2010 IEEE Network Operations and Management Symposium (NOMS), IEEE,* 2010, pp. 479-486.

[5] Y. Ben-Itzhak, I. Cidon, and A. Kolodny, "Performance and power aware cmp thread allocation modeling," *In HiPEAC*, 2010, pp. 232-46.

[6] S. Dawson-Haggerty, A. Krioukov, and D. E. Culler, "Power optimization - a reality check. *Technical Report UCB/EECS-2009-140*, EECS Department, University of California, Berkeley, Oct 2009.

[7] R. Urgaonkar, U.C. Kozat, K. Igarashi, and M.J. Neely, "Dynamic resource allocation and power management in virtualized data centers. *2010 IEEE Network Operations and Management Symposium (NOMS), IEEE*, 2010, pp. 479-486.

[8] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms. *In Proceedings of SIGMETRICS*, June 2009.

[9] S. Chatterjee and A. S. Hadi, Regression analysis by example. John Wiley and Sons, 2006.

[10] X. Fan, W.-D. Weber, and L.A. Barroso, "Power provisioning for a warehouse-sized computer.*In Proceedings of the 34th annual international symposium on Computer architecture*, San Diego, California, USA: ACM, 2007, pp. 13-23.

[11] R. V. der Wijngaart, NAS Parallel Benchmarks v. 2.4.NAS Technical Report NAS-02-007, October 2002.

[12] J. Subhlok, S. Venkataramaiah, and A. Singh, "Characterizing NAS Benchmark Performance on Shared Heterogeneous Networks," *International Parallel and Distributed Processing Symposium: IPDPS 2002 Workshops*, vol. 2, 2002, pp. 0086.

[13] http://wiki.xensource.com/xenwiki/xenpm. Accessed 30-6-2011.

[14] K. H. Kim, A. Beloglazov, and R. Buyya, "Power-aware provisioning of virtual machines for real-time Cloud services. Concurrency and Computation: Practice and Experience (2011), 23: 1491-1505.

[15] L. Bertini, J.C.B. Leite, and D. Mosse, "Statistical QoS Guarantee and Energy-Efficiency in Web Server Clusters," *In Proceedings of the 19th Euromicro Conference on Real-Time Systems. IEEE Computer Society*, 2007, pp. 83-92.

[16] J.S. Chase, D.C. Anderson, P.N. Thakar, A.M. Vahdat, and R.P. Doyle, "Managing energy and server resources in hosting centers," *In Proceedings of the eighteenth ACM symposium on Operating systems principles*, Banff, Alberta, Canada: ACM, 2001, pp. 103-116.

[17] E. N. Elnozahy, M. Kistler, and R. Rajamony, "Energy-efficient server clusters," *In Power-Aware Computer Systems, volume 2325 of Lecture Notes in Computer Science*, pp. 179-197, 2003. ISBN 978-3-540-01028-9.

[18] B. Khargharia, S. Hariri, and M.S. Yousif, "Autonomic power and performance management for computing systems," *Cluster Computing*, vol. 11, Jun. 2008, pp. 167-181.

[19] N. Nandola and S. Bhartiya, "A multiple model approach for predictive control of nonlinear hybrid systems," *Journal of Process Control*, vol. 18, 2008, pp. 131-148.

[20] D. Kusic and N. Kandasamy, "Power and performance management of virtualized computing environments via lookahead control," *In Proceedings of ICAC*, June 2009.

[21] X. Wang, D. Lan, X. Fang, M. Ye, and Y. Chen, "A resource management framework for multi-tier service delivery in autonomic virtualized environments," *In Proceedings of NOMS*, April 2008.

[22] S. Govindan, J. Choi, B. Urgaonkar, A. Sivasubramaniam, and A. Baldini, "Statistical profiling-based techniques for effective power provisioning in data centers," *In Proceedings of EuroSys*, April 2009.

[23] T. LI and L. K. JOHN, "Run-time modeling and estimation of operating system power consumption," *In Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, June 2003.

[24] P. Ranganathan and P. Leech, "Simulating complex enterprise workloads using utilization traces," *10th Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW)*, 2007.

[25] P. Ranganathan, P. Leech, D. Irwin, and J. Chase, "Ensemble-level Power Management for Dense Blade Servers," *In Proceedings of the 33rd annual international symposium on Computer Architecture, IEEE Computer Society*, 2006, pp. 66-77.

[26] T. Heath, B. Diniz, E.V. Carrera, J. Meira, and R. Bianchini, "Energy conservation in heterogeneous server clusters," *In Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming*, Chicago, IL, USA: ACM, 2005, pp. 186-195.

[27] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments," *Workshop on Modeling, Benchmarking, and Simulation (MoBS*), 2006.

[28] F. Bellosa, "The benefits of event: driven energy accounting in power-sensitive systems," *In Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, Kolding, Denmark: ACM, 2000, pp. 37-42.

[29] G. Contreras and M. Martonosi, "Power prediction for intel XScale R processors using performance monitoring unit events," *In Proceedings of the 2005 international symposium on Low power electronics and design*, San Diego, CA, USA: ACM, 2005, pp. 221-226.

[30] A. Jaiantilal, Y. Jiang, and S. Mishra, "Modeling CPU energy consumption for energy efficient scheduling," *In Proceedings of the 1st Workshop on Green Computing*, Bangalore, India: ACM, 2010, pp. 10-15.