A Probabilistic Model for Estimating the Power Consumption of Processors and Network Interface Cards

Waltenegus Dargie and Jianjun Wen
Chair for Computer Networks
Faculty of Computer Science
Technical University of Dresden
01062 Dresden, Germany
Email:waltenegus.dargie@tu-dresden.de, jianjun.wen@mailbox.tu-dresden.de

Abstract—**Many of the proposed mechanisms aiming to achieve energy-aware adaptations in server environments rely on the existence of models that estimate the power consumption of the server as well as its individual components. Most existing or proposed models employ performance (hardware) monitoring counters and the CPU utilization to estimate power consumption, but they do not take into account the statistics of the workload the server processes. In this paper we propose a lightweight probabilistic model that can be used to estimate the power consumption of the CPU, the network interface card (NIC), and the server as a whole. We tested the model's accuracy by executing custom-made benchmarks as well as standard benchmarks on two heterogeneous server platforms. The estimation error associated with our model is less than 1% for the custom-made benchmark whereas it is less than 12% for the standard benchmark.**

Index Terms—**Power consumption model, stochastic model, server power consumption, processor power consumption, NIC power consumption, probability distribution function, random variable**

I. Introduction

Studying the power consumption of large scale servers and data centers as a means to achieve energy-proportional computing is an active research area [3], [1], [17], [15]. Broadly speaking, the studies focus on one of the following aspects, namely, (1) on investigating the power consumption characteristics of a server as a whole or (2) on investigating the relationship between the power consumption and the workload of a server. The first study is useful for planning the power budget of a data center [31] and to design energy-efficient cooling systems [32]. Likewise, the second study is useful for various purposes such as achieving energy-aware workload placement [33], designing dynamic power management policies [34] and energy-aware task scheduling algorithms [22], and undertaking energy-aware service and workload consolidation [28], [27], [9], [29].

The models targeting the first aspect often aim to estimate the power consumption of an entire server or even an entire data center whereas those targeting the second aspect take a more fine-grained approach to estimate the power consumption of some of the individual subsystems of a server, such as a processor or a memory subsystem. Therefore, the former models aim to capture long term trends while the latter aim mainly to capture short term trends.

Among the models targeting the second aspect, many of them focus on the power consumption of the processor, since the processor is responsible for producing (when busy) the largest portion of the overall power consumption of a server [31]. Moreover, most of these models employ hardware performance counters (or performance monitoring counters) which provide a useful information about the activities of micro-architectural components inside the processor. The number and the types of performance monitoring counters each model selects depend on such factors as the architecture of the processor and the types of workloads the processor is expected to deal with. We will give a more detailed explanation on this subject in Section II.

In this paper, we propose a probabilistic model for estimating the power consumption of the processor and the network interface card of a server. We use one and the same approach for both subsystems and, complementary to hardware performance counters, employ the utilization statistics of the subsystem under consideration. Our approach can also estimate the power consumption of the entire server, but we do not consider it in this paper for lack of space.

The remaining part of this paper is organized as follows: In Section II, we review some of the proposed power estimation approaches. In Section III, we introduce our model and discuss its essential features in detail. In Section IV, we discuss the essential features of a memoryless system with a stochastic inputs – a prerequisite feature to apply our model. In Section V, we outline in detail our experiment setting and the methodology we adopt to measure the power consumption of the processor and the network interface card. In Section VI, we employ the power estimation model to estimate the power consumption of a processor for different types of workloads. Likewise, in Section VII, we apply the proposed model to estimate the power consumption of a network interface card. Finally, we give concluding remarks and outline future work in Section VIII.

II. Related Work

The power consumption of a server depends on both static and dynamic factors. Among the static factors are the type of hardware subsystems that make up the server and the efficiency of the software that manages these subsystems. The predominant dynamic factor is the workload of the server which is then reflected by the utilization level of the subsystems such as the CPU utilization, the memory utilization, the network bandwidth utilization, etc. Our focus is on the dynamic component of the power consumption and we use the term workload to refer to a utilization level.

Broadly speaking, the approaches pertaining to power consumption estimation can be classified into four different groups.

The first group attempts to establish the relationship between a known workload (for example, in terms of the number of requests per second, number of transactions per second, number of operations per second, etc.) and the overall power consumption of the entire system [23], [2], [29]. This approach simplifies the task because it is relatively easy to measure the AC power consumption. But it also includes the inefficiency of the power supply unit and the various voltage regulators into the estimation model. Moreover, it does not target the power consumption of the individual subsystems to understand the characteristic of the workload, which may be helpful for power-aware schedulers.

The second approach attempts to directly measure and relate the DC power consumption of the different subsystems (particularly, the processor, the memory, the network interface card, and the external storage devices) to their utilization level [21], [26], [7]. This approach, if successful, has two advantages. Firstly, it enables to apply separate dynamic power management policies to the individual subsystems. Secondly, it enables a power-aware scheduler to determine where to place a workload [30]. The two advantages are related to each other, but the second advantage is achieved by managing the workload (or the service) instead of the server. The difficulty with the second approach is that it is difficult to estimate the power consumption of the individual subsystems. As a result, it is inevitably made under several critical assumptions or by modifying the structure of the server to insert power meters. Our own approaches partially belongs to this group.

The third approach employs software simulation environments to estimate the power consumption of the individual hardware subsystems [14], [19], [24], [11]. Often the simulation environments take the peak power consumption of the various subsystems as a reference to establish the power model of a system. The difficultly with this approach is finding a mechanism to validate the accuracy of the estimation, since most hardware devices do not actually consume the power prescribed by the specification. It is also difficult to accommodate the power loss due to wear-and-tear and hardware inefficiencies.

The fourth approach, which is perhaps the most frequently used approach, employs hardware performance counters, assuming that the CPU is the predominant consumer of the dynamic component of the power consumption of a server [10], [25], [18], [6], [5]. A contemporary CPU provides one or more model-specific registers (MSR) that can be used to count certain micro-architectural events (or performance monitor events). The types of events that should be captured by a PMC is specified by a performance event selector (PES), which is also a MSR. The amount of countable events has been increasing with every generation, family, and model of processors. At present, a processor can provide more than 200 events. The motivation for using PMC is that accounting for certain events may offer detailed insight into the reason why the processor consumes power the way it does [4], [22]. PMC do not require the modification of or intrusion into the hardware structure. Moreover, the events they capture can accurately reflect the activity levels of the processor.

There are some challenges with employing hardware performance counters: Firstly, one is required to have knowledge of the low-level counters in order to be able to meaningfully correlate hardware events with the power consumption. Secondly, the identification of the relevant counters is strongly dependent on the nature of the benchmark and the server architecture. Thirdly, in most server architectures, one may be able to read not more than a few counters at the same time. This in turn may affect the analysis of the existence of correlation between the hardware events and the power consumption. We propose a light-weight stochastic model to estimate the probability distribution function of the power consumed by a hardware system as long as this system can be modeled as a memoryless system with a stochastic input. This assumption can be satisfied if the power consumption and the utilization of the hardware system are sampled at an appropriate granularity (in the range of seconds). The only input our model requires is the statistics of the utilization. We will demonstrate the scope and usefulness of our model by estimating the power consumption of a processor and a network interface card of two different servers.

III. Stochastic Model

Before we begin with the introduction of our model, we explain how we represent variables. A boldface lower case letter ($\mathbf{w}$) refers to a random variable. A normal lower case letter (w) refers to a real number associated with the random variable $\mathbf{w}$. An upper case F refers to the cumulative distribution function[1] (CDF) while a lower case f refers to the probability density function[2].

A. Known Relationship

Suppose we wish to reason about the relationship between the utilization of a system (for example, the utilization of a processor) $\mathbf{w}$ and its power consumption $\mathbf{p}$. Modeling these two quantities as random variables or random processes is reasonable because they cannot be known in advance except in a probabilistic sense. Consequently, one way to reason about their relation is to observe and examine their statistics. We assert that if the subsystem can be considered as a memoryless system and if the relationship between $\mathbf{w}$ and $\mathbf{p}$ can be represented by a one-to-one function, then examining the cumulative distribution functions of the two quantities can be sufficient to establish a quantitative relationship between them.

To highlight our point, we shall begin by assuming that the relationship is already known. Hence, we wish to determine the CDF of one of the random variables (the one whose statistics we do not know) in terms of the the other (whose statistics we do know). For example, if the power consumption of a processor is expressed as:

$$\mathbf{p} = a\mathbf{w} + b \quad a, b > 0. \tag{1}$$

Then, $F_p(p) = P\{\mathbf{p} \leq p\} = P\{a\mathbf{w} + b \leq p\} = P\left\{\mathbf{w} \leq \left[\frac{p-b}{a}\right]\right\} = F_w\left(\left[\frac{p-b}{a}\right]\right)$, where $F_w(p)$ refers to the distribution of $\mathbf{w}$ expressed in terms of p. Likewise, the probability density function of $\mathbf{p}$ can be expressed as

[1] The distribution function $F_w(w)$ of the random variable $\mathbf{w}$ is defined as $F_w(w) = P\{\mathbf{w} \leq w\}$, for $-\infty \leq w \leq \infty$. The distribution function is a non-decreasing, right continuous function, i.e., if w_1 and w_2 are two real numbers and $w_2 > w_1$, then $F_w(w_2) \geq F_w(w_1)$, $\forall w_2, w_1$.

[2] The probability density function of $\mathbf{w}$ is the derivation with respect to w of $F_w(w)$, $f(w) = \frac{dF(w)}{dw}$.

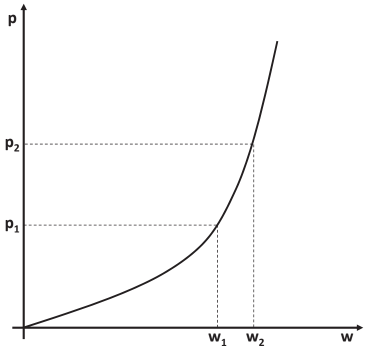

Fig. 1. Exploiting the one-to-one relationship between **p** and **w** and the monotonic nature of distribution functions to determine a quantitative relationship between **w** and **p**.

$\frac{d}{dp}\left(F_p(p)\right) = \frac{1}{a} f_w\left(\left[\frac{p-b}{a}\right]\right)$, where $f_w(p)$ refers to the density of **w** expressed in terms of p. If, for instance, the density of **w** is exponential: $f(w) = \lambda e^{-\lambda w}$, $\lambda > 0$, where λ is the inverse of the mean of the random variable, then, $f_p(p) = \frac{\lambda}{a} e^{-\lambda\left(\left[\frac{p-b}{a}\right]\right)}$.

B. Unknown Relationship

If, however, the relationship between **w** and **p** is not known (which is why we need to develop a model), then the task can be considered as the inverse process of Section III-A. Hence, given two distribution functions $F_w(w)$ and $F_p(p)$ which we know are related to each other, our task is to determine the exact nature of the relationship. In other words, we provide the system a workload of known statistics and observe the statistics of the power consumption of the system. Then we should find a function $g(\mathbf{w})$ such that the distribution of $\mathbf{p} = g(\mathbf{w})$ equals $F_p(p)$.

If the relationship between **p** and **w** can be estimated as a one-to-one function, i.e., every element of the range of **p** corresponds to exactly one element of the domain of **w**[3], then $P\left\{\mathbf{p} \le p_i\right\}$ equals to $P\left\{\mathbf{w} \le w_i\right\}$ because $\mathbf{p} \le p_i$ if and only if $\mathbf{w} \le w_i$. This can be better visualized in Figure 1 which displays a one-to-one function. From the figure it is apparent that the value p_2 corresponds to w_2. Therefore, $P\left\{\mathbf{p} \le p_2\right\}$ corresponds to $P\left\{\mathbf{w} \le w_2\right\}$. Similarly, $P\left\{\mathbf{p} \le p_1\right\}$ corresponds to $P\left\{\mathbf{w} \le \mathbf{w}_1\right\}$. From this, we can conclude that for a one-to-one function:

$$F_p(p_i) = P\left\{\mathbf{p} \le p_i\right\} = P\left\{\mathbf{w} \le w_i\right\} = F_w(w_i) \qquad (2)$$

Subsequently, using Equation 2, we can express **p** in terms of $F_w(p)$ and $F_w(w)$ as follows:

$$p_i = F_P^{-1}(F_w(w_i)) \qquad (3)$$

where F_p^{-1} refers to the inverse of $F_p(p)$ [35]. For example, if we observe a uniformly distributed power consumption in the rage of (10, 50) W for an exponentially distributed workload, $F_w(w) = 1 - e^{-\lambda w}, \lambda > 0$, then, using Equation 2 we have: $\left(1 - e^{-\lambda w}\right) = \frac{1}{40} p$, from which, $p = 40(1 - e^{-\lambda w}) = 40 F_w(w)$.

IV. Memoryless Property

Equation 3 is a useful expression, but its usefulness is bound to two conditions, namely, (1) the system is memoryless and (2) the workloads should be statistically stationary. Fulfilling the second condition is possible, since the stochastic property of the workload can be controlled during the experiment. Fulfilling the second condition requires choosing the appropriate measurement granularity – if the sampling interval is fine grained, the system may not be considered as a memoryless system, because there can be a strong dependency between the samples of the measurement. This is particularly true of processors. If, on the other hand, there is a sufficient distance between the samples of the power consumption and the utilization (for the processors we considered, a sampling interval in the range of hundred milliseconds was sufficient), then the dependency between the samples becomes weak and the system can be regarded as memoryless.

The autocorrelation function, $R_{ww}(t_2, t_1) = E[w(t_2)w(t_1)]$ [35], is the best tool to measure the degree of dependency between the samples of a stationary random process, $w(t)$. The difference between t_2 and t_1 refers to the time lag between the samples. For a memoryless system, $R_{pp}(t_2, t_1) \approx 0$ for $t_2 \neq t_1$ if $R_{ww}(t_2, t_1) \approx 0$ for $t_2 \neq t_1$. $R_{pp}(t_2, t_1)$ and $R_{ww}(t_2, t_1)$ are the autocorrelation functions of **p** and **w**, respectively.

Figure 2 displays the autocorrelation functions of uniformly distributed CPU and NIC workloads (utilization). For both devices the utilization was sampled every second. The autocorrelation of the CPU utilization displays the existence of an apparent correlation for a time lag less than eight seconds, because we deliberately introduced dependency between the samples of the eight consecutive seconds (we refer the reader to Section 5 to learn how the CPU workload was generated). For a time lag of greater than eight seconds, however, the autocorrelation drops nearly to zero. Likewise, the autocorrelation of the NIC workload drops sharply for $t = t_2 - t_1 > 0$. These observation confirm that for a sampling interval of one second or even a few hundred milliseconds, our assumption that the two systems can be modeled as memoryless systems with stochastic inputs is plausible.

Figure 3 shows the autocorrelation functions of the corresponding power consumptions of the CPU and the NIC. Unlike the workloads which were sampled every second, we sampled the power consumption every 250 ms on average because of the relatively high resolution of the devices we used to measure power (Yokogawa WT210 digital power analyzers). Therefore the time lag 240 in Figure 3 corresponds to the 60 s time lag in Figure 2. With this adjustment and taking into account how we generated the workload of the CPU, it is clear that the correlation between the samples becomes weak for a time lag greater than 32 (corresponding to 8 s), confirming our assumption that the processor can indeed be regarded as a memoryless system. The autocorrelation of the power consumption of the NIC dropped to zero for $t_2 - t_1 > 0$, since all the samples of the workload were independent.

V. Experiment Setting

In this section, we explain how we applied the concepts we developed in the previous sections to estimate the power consumption of a processor and a network interface card (NIC) based on their utilization statistics.

We performed our experiment on two heterogeneous server platforms. The first one was built on a D2581 Siemens-Fujitsu

[3] For example, the function $\mathbf{p} = a\mathbf{w} + b$ is a one-to-one function, since **p** has exactly one solution $\forall w$. Likewise, $\mathbf{p} = a\mathbf{w}^2 + b$ has a single solution for $w > 0$.

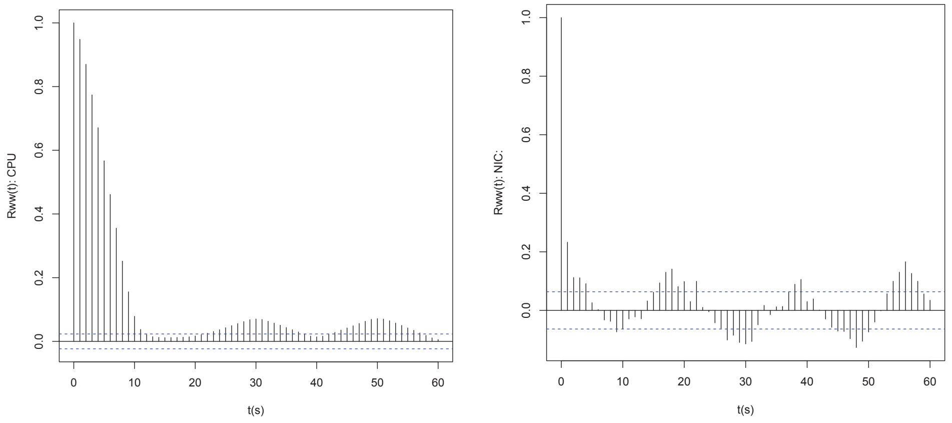

Fig. 2. The autocorrelation function of uniformly distributed CPU (top) and NIC (bottom) utilization. Maximum time lag $t = (t_2 - t_1) = 60$ seconds.

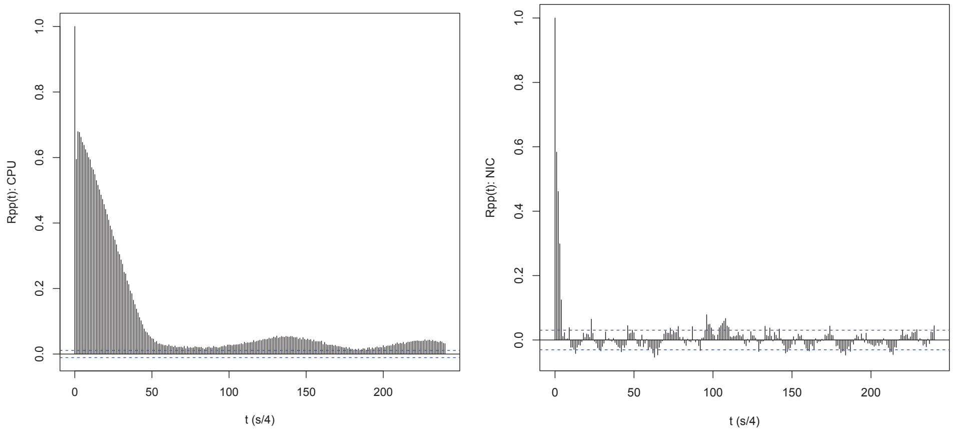

Fig. 3. The autocorrelation function of the power consumption of the CPU (left) and the NIC (right) for uniformly distributed workloads. Maximum time lag $t = (t_2 - t_1) = 60$ seconds.

motherboard integrating a 3.16 GHz Intel E8500 dual core processor. The second server was built on a DB65Al Intel motherboard integrating a 2.5 GHz Intel i5-2400S dual core processor and a 1 GBit Intel network interface card. For testing the processor power model we used the E8500 server and for testing the NIC model, we used the i5-2400S server.

The motherboards of both servers provide two DC power connectors to supply the various subsystems with power. One of them is a 12 V, 4-pole connector whereas the other is a 24-pole connector with 12 V, 5 V and 3.3 V rails (among others). The 12 V rail of the 4-pole connector is exclusively used by the voltage regulators of the processor in both motherboards to generate the core voltages. The voltage regulators draw some amount of power from the 5V rail of the 24-pole connector mainly used by the Pulse Width Modulator controllers and it is comparatively very small. The 3.3 V of the 24-pole connector is predominantly used by the Low Pin Count (LPC) IO controllers. Devices connected to the PCI and PCI Express cards, such as the network interface card, exclusively draw power through the 3.3 V rail.

For establishing the relationship between the power consumption and the utilization of the processor and the NIC, we generated CPU-bound and IO-bound workloads having different resource utilization characteristics and executed them on the two servers. The CPU-bound workload was a convolution operation in which integer, float point, and shift operations were performed. While the convolution operation was executed, it utilised 100% of the CPU, but when the loop operation was not executed, the CPU was idle. In order to generate the desired workload distribution, we divided time into a sequence of one-second none overlapping windows. We then generated a set of random numbers in the interval [0, 100] using the `runif` function of the `R` statistical tool to make sure that the distribution of the random numbers is uniform. In each time window, we picked out one of these random numbers to determine the portion of time the CPU was fully utilized by the convolution operation (between 0 and 100% of the time window).

In order to avoid instability in computation, the proportion of the CPU utilization for the subsequent eight windows was made the same. This means that there was an apparent correlation between the eight consecutive windows; otherwise, the random numbers we picked out were independent. The program run for one hour. For testing the model, we generated an exponentially distributed convolution operation in addition to the SPEC Power 2008 standard benchmark provided by

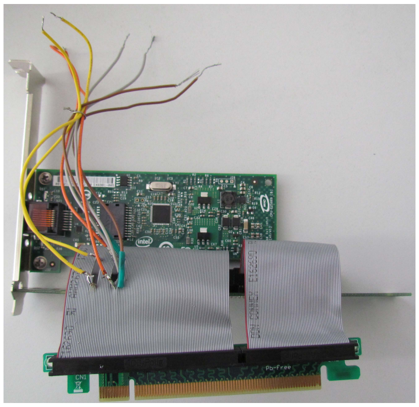

Fig. 4. Using a raiser board to intercept the power rails of the PCI Express to measure the power consumption of the NIC.

the Standard Performance Evaluation Corporation[4]. We tested our model while the server run under three different dynamic frequency and voltage scaling policies.

To establish the relationship between $\mathbf{w}$ and $\mathbf{p}$ of the NIC, we followed the same approach, but this time, instead of the convolution operation, we used an application that uploads data on the i5-2400S server at different transmission rates thus varying the amount of network bandwidth utilized per second (MBps) according to a predefined probability distribution function, namely, uniform and exponential distributions. The application run for 15 minutes for each configuration. To measure the power consumption of the NIC, we connected it with a raiser card which is in turn connected to the PCI Express bus. We intercepted the 3.3 V rail of the raiser board to directly measure the power drawn by the NIC. Figure 4 displays the NIC instrumentation.

VI. Processor Power Consumption Model

To determine the relationship between the utilization statistics of the E8500 Intel processor and its power consumption, we first disabled one of the cores[5]. Then, we generated a one hour uniformly distributed workload in the interval (0,100) by the convolution operation. We choose a uniformly distributed workload because it simplifies the evaluation of Equation 3. During the execution of the convolution operation, we measured the power consumption of the processor and plotted its $F_p(p)$. Then, using R's `nls` curve fitting toolbox, we approximated $F_p(p)$ which can be best approximated by a quadratic function, $F_p(p) = a_1 p^2 + a_2 p + a_3, 9.45 \le p \le 32.15$, where $a_1 = -0.001026$, $a_2 = 0.07765$, and $a_3 = -0.6136$ are the coefficients of the quadratic function. Figure 5 displays the experimental and the approximated $F_p(p)$. Hence, for the quadratic functions, we have:

$$F_p(p) = a_1 p^2 + a_2 p + a_3 \quad 9.54 \le p \le 32.15 \tag{4}$$

$$F_w(w) = \frac{w}{100}, \quad 0 \le \mathbf{w} \le 100 \tag{5}$$

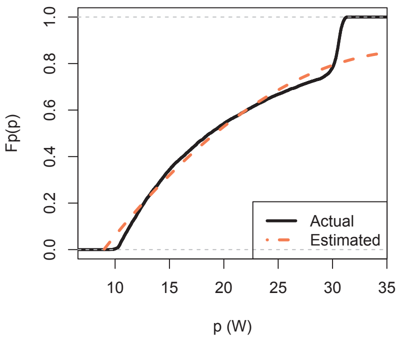

Fig. 5. The actual (measured) and estimated values of $F_p(p)$ of the uniformly distributed utilization $U(10, 90)$ for the Intel E8500 dual core processor.

By inserting Equation 4 and 5 into Equation 3, we obtain:

$$\mathbf{p} = \frac{-a_2 + \sqrt{{a_2}^2 - 4a_1 \times (a_3 - \frac{\mathbf{w}}{100})}}{2 \times a_1} \tag{6}$$

Equation 6 can be expressed as:

$$\mathbf{p} = K_1 + (K_2\mathbf{w} + K_3)^{\frac{1}{2}} \tag{7}$$

where $K_1 = -\frac{a_2}{2a_1}$, $K_2 = \frac{1}{a_1(100)}$, and $K_3 = \frac{a_2^2}{4a_1} - \frac{b}{a_1(100)} - \frac{a_3}{a_1}$.

Equation 7 is the desired relationship we wished to establish between the CPU workload and the power consumption. Using this relationship, it is now possible to estimate the runtime power consumption of the processor as long as we can predict its utilization. Moreover, using Equation 7, we can determine the distribution and density of $\mathbf{p}$ for a workload of arbitrary distribution and density.

Earlier, we showed that $F_p(p)$ can be expressed as $P\{\mathbf{p} \le p\} = P\{g(\mathbf{w}) \le p\}$. Hence,

$$F_p(p) = P\left\{\left(K_1 + (K_2\mathbf{w} + K_3)^{\frac{1}{2}}\right) \le p\right\} \quad 0 \le \mathbf{w} \le 100 \tag{8}$$

Expressing Equation 8 in terms of $F_w(w)$ yields:

$$F_w(p) = P\left\{\mathbf{w} \le \frac{(p - K_1)^2 - K_3}{K2}\right\} \quad b \le \mathbf{w} \le c \tag{9}$$

which is the same as $F_w\left(\frac{(p-K_1)^2 - K_3}{K2}\right)$. Likewise, the density of $\mathbf{p}$ can be expressed as:

$$f_p(p) = \left|\frac{2}{K_2}(p - K_1)\right| f_w\left(\frac{(p - K_1)^2 - K_3}{K2}\right) \tag{10}$$

A. Theoretical $F_p(p)$ and $f_p(p)$

Using the relationship expressed in Equation 7, it is possible to compute the distribution and density functions of $\mathbf{p}$ for a workload of arbitrary probability density function. We shall demonstrate this by computing the theoretical density and distribution functions of $\mathbf{p}$ for an exponentially distributed workload. In the subsection that will follow we shall compare

[4] www.spec.org/power_ssj2008/

[5] The relationship between the workload and the power consumption of the dual-core processor is linear: $\mathbf{p} = K_4\mathbf{w} + K_5$ where $K_4 = 3.5$ and $K_5 = 10.5$, $0 \le \mathbf{w} \le 100$. The derivation of this relationship is beyond the scope of this paper.

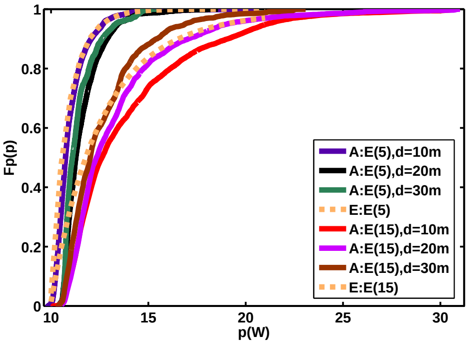

Fig. 6. The actual (A) and estimated (E) $F(p)$ of the Intel E8500 processor when executing exponentially distributed workloads.

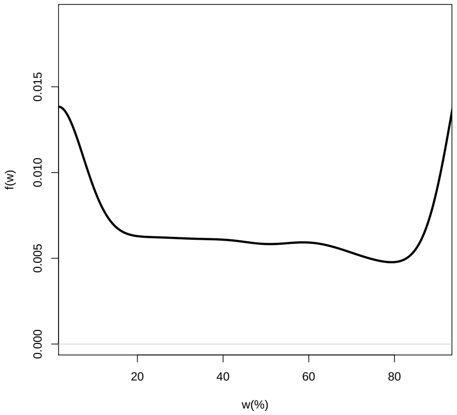

Fig. 7. The actual workload distribution of the SPEC power benchmark.

the theoretical result with the one we obtained from an experiment.

1) Exponentially distributed workload: When **w** is exponentially distributed ($f(w) = \lambda e^{-\lambda w}; \mu = \frac{1}{\lambda}$), its distribution function equals:

$$F_p(w) = 1 - e^{-\frac{w}{\mu}} \quad b \leq w \leq c \tag{11}$$

And the probability density function of **p** can be expressed as follows:

$$f_p(p) = \left| \frac{2}{K_2}(p - K_1) \right| e^{-\left(\left((p-K_1)^2 - K_3\right)/K_2\right)/\mu} \tag{12}$$

The distribution function of **p** is expressed as:

$$F_p(p) = F_w(p) = 1 - e^{-\left(\left((p-K_1)^2 - K_3\right)/K_2\right)/\mu} \tag{13}$$

where p is in the interval [9.45, 32.15] and $F_w(p)$ refers to the probability distribution function of **w** expressed in terms of **p**.

B. Experimental $F(p)$

After having established the relationship between **p** and **w**, we tested the validity of our model by generating custom-made exponentially distributed workloads and the SPEC Power benchmark. The workloads with the exponential distribution had the following average utilization: $\mu = 5\%, 10\%, 15\%$ and 20%. To ensure that our model was time invariant, we generated the workloads for the duration of 10, 20, and 30 minutes (i.e., the sample size of the workload for each test case was different). Figure 6 displays the theoretically estimated (the dashed lines) and the experimentally obtained (solid lines) $F_p(p)$ for the exponentially distributed workloads of the E8500 processor. As can be seen from the figure, when the test workload was similar in type with the training workload (in both cases we used the convolution operation), its power consumption could be accurately predicted (with an average error $< 1\%$) even though the statistics of the workloads were dissimilar and the durations of the workloads were different for the test cases.

Similarly, we tested our model with the SPECpower_ssj2008 (SPEC Power) benchmark. The SPEC Power "is the first industry-standard SPEC benchmark that evaluates the power and performance characteristics of volume server class and multi-node class computers"[6]. The full SPEC power benchmark runs for 70 minutes. We tested the model while the server operated under different dynamic voltage and frequency scaling policies, to examine how its estimation accuracy was affected by frequency and voltage variations (real-world servers often employ dynamic voltage and frequency scaling for energy-efficient operation). The policies we examined were the *performance*, *conservative*, and *on-demand* policies. For the detail of these policies, we refer our reader to our previous work [13, 8]. The conservative and on-demand policies vary the frequency and core voltage of the processor depending on its anticipated future workload using exponential moving average filters to predict the future workload of the processor. Figure 8 displays the probability density functions of the actual and estimated power consumed by the E8500 processor when executing the SPEC Power benchmark. Unlike the exponentially distributed workload, the SPEC Power benchmark occupies the entire utilization domain (see Figure 7) which in turn resulted in a wider power consumption range.

It must be noted that the model parameters of Equation 7 were obtained when the server was operating at maximum core voltage and maximum operation frequency whereas when we tested the model, the processor operation voltages and frequencies were dynamically varied by the power management policies. Even so, our model was able to estimate the power consumption of the processor in all the power management settings with comparable accuracy. The average estimation error was 11.3%.

VII. NIC Power Model

Similar to Section VI, we employed Equations 2 and 3 to estimate the power consumption of the network interface card of the i5-2400S server. We shall show how the relationship between **w** and **p** was determined for a uniformly distributed bandwidth utilization and how, using the relationship, we estimated the power consumption statistic of the NIC for a utilization of arbitrary statistics.

[6] http://www.spec.org/power_ssj2008/

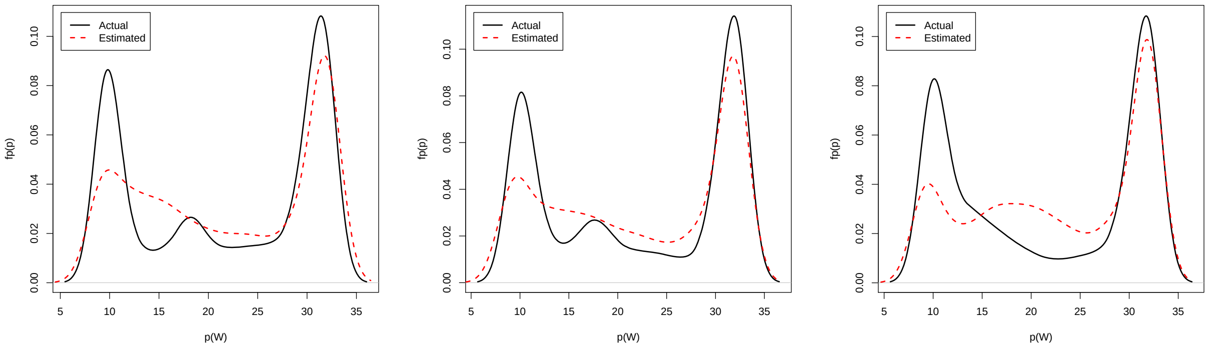

Fig. 8. The probability density function of the actual and estimated power consumption of the processor for the SPEC Power benchmark. The processor was running under the performance (left), conservative (middle), and on-demand (right) dynamic voltage and frequency scaling policies.

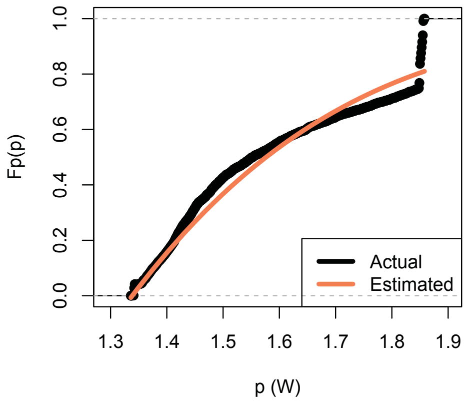

Fig. 9. The actual (measured) and estimated values of $F_p(p)$ of the uniformly distributed bandwidth utilization for the i5-2400S server.

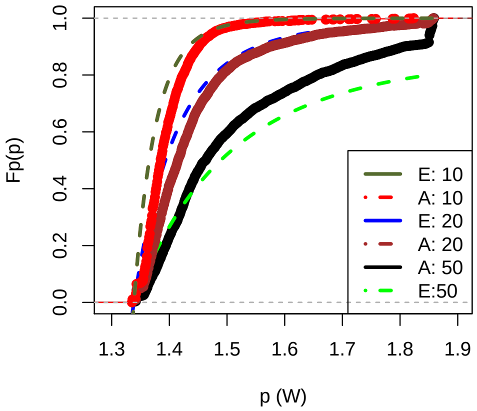

Fig. 10. The actual (measured) and estimated values of $F_p(p)$ of the exponentially distributed bandwidth utilization for the i5-2400S server.

A. Model Parameters

Figure 9 (the black solid line) displays the distribution of the actual power consumption of the NIC when its 15 minute bandwidth utilization was uniformly distributed in the interval [0, 125] MBps. We approximated this distribution using a curve fitting by the following expression (the coral solid line in Figure 9):

$$F_p(p) = 1 - 6e^{-p^2} \tag{14}$$

Since the network bandwidth utilization varied uniformly, we have $F_w(w) = \frac{w}{125}$ for $0 \leq w \leq 125$ MBps. Thus using the inverse relation we obtained in Equation 3, the power consumption of the network interface card can be expressed as follows:

$$p = \sqrt{-ln\left[\frac{125 - w}{750}\right]} \tag{15}$$

And the probability distribution function of $\mathbf{p}$ in terms of the distribution function of $\mathbf{w}$ is expressed as:

$$F_p(p) = F_w\left(1 - 6e^{-p^2}\right) \tag{16}$$

Finally, the density of $\mathbf{p}$ is expressed as:

$$f(p) = 12pe^{-p^2} f_w\left(125 - 600e^{-p^2}\right) \tag{17}$$

B. Model Validation

Figure 10 displays the power consumption of the NIC when its bandwidth utilization was exponentially distributed. We have considered three cases: $\mu = 10, 20, 50$. Except for the case $\mu = 50$, the relationship in Equation 15 was able to accurately estimate the power consumption of the NIC based on knowledge of the statistics of the bandwidth utilization (with a standard error that equals 0.00955). The estimation error was larger for $\mu = 50$ (a standard error of 0.0146). The justification for this is similar to the one we gave to the SPEC Power benchmark – as the span of the utilization domain increased, the accumulated estimation error increased as well. Even so, the average estimation error, similar to the estimation error of the processor for the exponential workload, was $< 1\%$.

VIII. Conclusion

We proposed a probabilistic model to estimate the power consumption of the different subsystems of a server. Our model uses two variables only, namely, the system's utilization statistics ($\mathbf{w}$) and the statistics of the actual power consumption ($\mathbf{p}$), both are easily obtainable in many server platforms. In our model the cumulative distribution function played a vital role.

We demonstrated the scope and usefulness of our approach by estimating the power consumption of the processor and the network interface card of two heterogeneous servers for a variety of workload statistics. Altogether we executed 16 test cases. This said, the model is useful for memoryless systems, which means, the sampling interval should be long enough to ensure that the samples of the workload and the power consumption should be statistically independent.

Our model performed relatively poorly for the SPECPower benchmark. This is because, the estimation region occupies the entire utilization domain of the processor thereby increasing the accumulated estimation error. A typical Internet server may not have such a wide utilization domain. One of the problems we faced during the testing of our model was the difficulty of using curve fitting. Without this step, it was not possible to establish a relationship between **w** and **p**. For a curve fitting to produce an accurate approximation, the expressions should be complex. Simple expressions come up with large errors. But obtaining the inverse of complex expressions is difficult.

In this paper we have not included the memory subsystem, without which it is difficult to estimate the overall DC power consumption of a server. Our aim in future is to include it and to compute the contribution of individual components to the overall power consumption of a server.

References

[1] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu. Energy proportional datacenter networks. SIGARCH Comput. Archit. News, 38(3):338347, June 2010.

[2] F. Ahmad and T. N. Vijaykumar. Joint optimization of idle and cooling power in data centers while maintaining response time. In Proceedings of the fifteenth edition of ASPLOS on Architectural support for programming languages and operating systems, ASPLOS 10, pages 243256, New York, NY, USA, 2010. ACM.

[3] L. A. Barroso and U. Holzle. The case for energy-proportional computing. Computer, 40(12):3337, Dec. 2007.

[4] F. Bellosa. The benefits of event. In Proceedings of the 9th workshop on ACM SIGOPS European workshop beyond the PC: new challenges for the operating system - EW 9, page 37, New York, New York, USA, 2000. ACM Press.

[5] R. Bertran, M. Gonzalez, X. Martorell, N. Navarro, and E. Ayguade. Decomposable and responsive power models for multicore processors using performance counters. In Proceedings of the 24th ACM International Conference on Supercomputing - ICS 10, page 147, New York, New York, USA, 2010. ACM Press.

[6] W. L. Bircher and L. K. John. Complete System Power Estimation Using Processor Performance Events. IEEE Transactions on Computers, 61(4):563577, Apr. 2012.

[7] P. Bohrer, E. N. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, C. McDowell, and R. Rajamony. The case for power management in web servers, pages 261289. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[8] A. Brihi and W. Dargie. Dynamic voltage and frequency scaling in multimedia servers. In The 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013), 2013.

[9] G. Chen,W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, NSDI08, pages 337350, Berkeley, CA, USA, 2008. USENIX Association.

[10] X. Chen, C. Xu, R. P. Dick, and Z. M. Mao. Performance and power modeling in a multi-programmed multi-core environment. In Proceedings of the 47th Design Automation Conference, DAC 10, pages 813818, New York, NY, USA, 2010. ACM.

[11] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. SIGMETRICS Perform. Eval. Rev., 33:303314, June 2005.

[12] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2, NSDI05, pages 273286, Berkeley, CA, USA, 2005. USENIX Association.

[13] W. Dargie. Analysis of the power consumption of a multimedia server under different DVFS policies. In IEEE CLOUD, 2012.

[14] D. Economou, S. Rivoire, and C. Kozyrakis. Full-system power analysis and modeling for server environments. In The 2nd Workshop on Modeling, Benchmarking, and Simulation (MoBS), pages 7077, 2006.

[15] E. N. Elnozahy, M. Kistler, and R. Rajamony. Energy-efficient server clusters. In Proceedings of the 2nd international conference on Power-aware computer systems, PACS02, pages 179197, Berlin, Heidelberg, 2003. Springer-Verlag.

[16] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper. Resource pool management: Reactive versus proactive or lets be friends. Comput. Netw., 53:29052922, December 2009.

[17] S. Huang and W. Feng. Energy-efficient cluster computing via accurate workload characterization. In Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 09, pages 6875, Washington, DC, USA, 2009. IEEE Computer Society.

[18] 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, HotPower08, pages 44, Berkeley, CA, USA, 2008. USENIX Association.

[19] C.-H. Lien, Y.-W. Bai, and M.-B. Lin. Estimation by software for the power consumption of streaming-media servers. IEEE T. Instrumentation and Measurement, 56(5):18591870, 2007.

[20] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In INFOCOM, pages 10981106, 2011.

[21] A. Mahesri and V. Vardhan. Power consumption breakdown on a modern laptop. In (PACS04), pages 165180, 2004.

[22] A. Merkel and F. Bellosa. Balancing power consumption in multiprocessor systems. ACM SIGOPS Operating Systems Review, 40(4):403, Oct. 2006.

[23] R. Nathuji and K. Schwan. Virtualpower: Coordinated power management in virtualized enterprise systems. In 21st ACM Symposium on Operating Systems Principles (SOSPS07), 2007.

[24] M. Poess and R. O. Nambiar. Energy cost, the key challenge of todays data centers: a power consumption analysis of tpc-c results. Proc. VLDB Endow., 1:12291240, August 2008.

[25] K. Singh, M. Bhadauria, and S. A. McKee. Real time power estimation and thread scheduling via performance counters. SIGARCH Comput. Archit. News, 37(2):4655, July 2009.

[26] D. C. Snowdon, S. M. Petters, and G. Heiser. Accurate on-line prediction of processor and memory energy usage under voltage scaling. In Proceedings of the 7th ACM & IEEE international conference on Embedded software, EMSOFT 07, pages 8493, New York, NY, USA, 2007. ACM.

[27] S. Srikantaiah, A. Kansal, and F. Zhao. Energy aware consolidation for cloud computing. In Proceedings of the 2008 conference on Power aware computing and systems, HotPower08, pages 1010, Berkeley, CA, USA, 2008. USENIX Association.

[28] A. Verma, G. Dasgupta, T. K. Nayak, P. De, and R. Kothari. Server workload analysis for power minimization using consolidation. In Proceedings of the 2009 conference on USENIX Annual technical conference, USENIX09, pages 2828, Berkeley, CA, USA, 2009. USENIX Association.

[29] Q. Zhu, J. Zhu, and G. Agrawal. Power-aware consolidation of scientific workflows in virtualized environments. In Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC 10, pages 112, Washington, DC, USA, 2010. IEEE Computer Society.

[30] J. Zhuo and C. Chakrabarti. Energy-efficient dynamic task scheduling algorithms for DVFS systems. ACM Trans. Embed. Comput. Syst., 7(2):17:117:25, Jan. 2008.

[31] X. Fan, W.-D. Weber, and L.A. Barroso. Power provisioning for a warehouse-sized computer. SIGARCH Comput. Archit. News 35, 2 (June 2007), 13-23.

[32] A. Beloglazov and R. Buyya. Energy efficient resource management in virtualized cloud datacenters. Presented at 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid). 2010.

[33] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making scheduling ”cool”: temperature-aware workload placement in data centers. In Proceedings of the annual conference on USENIX Annual Technical Conference (ATEC '05). USENIX Association, Berkeley, CA, USA, 5-5.

[34] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi. An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget. In Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 39, pages 347358, Washington, DC, USA, 2006. IEEE Computer Society.

[35] A. Papoulis and S. U. Pillai. Probability, random variables, and stochastic processes. McGraw Hill, 4th edition, 2002.