

# A Stochastic Model for Estimating the Power Consumption of a Processor

Waltenegus Dargie, *Senior Member, IEEE*

**Abstract**—Quantitatively estimating the relationship between the workload and the corresponding power consumption of a multicore processor is an essential step towards achieving energy proportional computing. Most existing and proposed approaches use Performance Monitoring Counters (Hardware Monitoring Counters) for this task. In this paper we propose a complementary approach that employs the statistics of CPU utilization (workload) only. Hence, we model the workload and the power consumption of a multicore processor as random variables and exploit the monotonicity property of their distribution functions to establish a quantitative relationship between the random variables. We will show that for a single-core processor the relationship is best approximated by a quadratic function whereas for a dualcore processor, the relationship is best approximated by a linear function. We will demonstrate the plausibility of our approach by estimating the power consumption of both custom-made and standard benchmarks (namely, the SPEC power benchmark and the Apache benchmarking tool) for an Intel and AMD processors.

**Index Terms**—DC power consumption, processor power, power model, multicore processor, processor workload analysis, stochastic model, processor power consumption estimation

## 1 INTRODUCTION

THE power consumption of multicore processors has been a subject of extensive research. For example, it has been studied (1) to understand the power density characteristic of chip microprocessing [1], (2) to design realistic and efficient cooling systems [2], (3) to develop energy- and thermal-aware schedulers [3], [4], (4) to support runtime task migration [5]; (5) to implement dynamic power management policies (dynamic voltage and frequency scaling) [6], and (6) to quantify the energy consumed by a piece of software [7] or an application [8]. More recently, it has become a subject of research in cloud and high performance computing to examine the proportionality between the power consumed by servers and the work they accomplish [9], [10], [11], [12].

As a result, a substantial body of work exists on power consumption estimation models for multicore processors [13]. In most cases events emitted by hardware performance counters (or performance monitoring counters (PMC)) are employed because they provide a useful insight into the activities of the various micro-architectural components of a processor. The number and types of events that should be captured depend on such factors as the type of workloads (benchmarks) used to train and test the model, the architecture of the target processor cores, and the desired granularity of the model. Often the level of correlation between the events and the power consumption of the processor is studied to select the best representative

counters. There are also models which combine CPU utilization information with hardware events because the former is easy to obtain [13], [14].

Regardless of the types of inputs, most existing models focus on estimating the instantaneous power consumption of a processor. While this approach enables a direct translation of the CPU activity level into power consumption level, it does not provide comprehensive knowledge about the power consumption *characteristic* of the processor. Similarly, it does not reveal sufficient insight into the *characteristic* of the workload which induces the power consumption. Understanding these two characteristics is useful for runtime adaptation such as energy-aware task scheduling, load-balancing, or workload consolidation. More importantly, the existing models consider the relationship between the workload and the power consumption of a processor as deterministic and invariable, which is not the case in reality. In contrast, a probabilistic approach can accommodate imprecise relationship and uncertainty about the input.

In this paper, we propose a stochastic power consumption estimation model for a multicore processor. In addition to the instantaneous power consumption, it estimates the statistics (CDF and pdf) of the power consumed by a processor and relates it to the statistics of its workload. Our approach has three useful purposes: (1) It enables a scheduler to estimate the power budget of a future workload, so that it can decide where to execute it; (2) it captures the magnitude and frequency of power fluctuations; and (3) it can estimate the probability that a workload execution crosses a certain power threshold (budget) that may result in overloading or underutilizing the processor.

The remaining part of this paper is organized as follows: In Section 2, we summarize the approaches similar to ours and outline how our approach differs from them. In Section 3, we justify the rationale for the selection of the model's input. In Section 4, we describe in detail the experiment setting for

• The author is with the Chair of Computer Networks, Faculty of Computer Science, Technical University of Dresden, 01062 Dresden, Germany.  
E-mail: waltenegus.dargie@tu-dresden.de.

Manuscript received 1 Jul. 2013; revised 6 Jan. 2014; accepted 25 Mar. 2014; date of publication xx xx xxxx; date of current version xx xx xxxx.

Recommended for acceptance by R. G. Melhem.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TC.2014.2315629

our approach. In Section 5, we develop an abstract stochastic model to establish the relationship between a processor's workload and its power consumption. In Section 6, we further develop the stochastic model to estimate the power consumption of a single-core processor. In Section 7, we extend the stochastic model to estimate the power consumption of a multicore processor. In Section 8, we provide a quantitative account of the model's estimation error and in Section 9, we compare our model with some of the proposed models. Finally, in Section 10, we provide concluding remarks.

## 2 RELATED WORK

Estimation of the power consumption of a processor due to a piece of workload involves measuring the actual power consumption and then relating it to some observable parameters that reflect the activity or the utilization of the processor. The idea is to establish a quantitative relationship between the observable parameters and the power consumption, so that the estimation task can bypass the need to actually measure the power consumption (which may involve intrusive and expensive instrumentation).

Most existing models employ performance monitoring counters along with micro benchmarks. 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, PME). For example, such an event will be generated when the processor retires (finishes) an instruction or when it accesses the cache. 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 power consumption characteristics of the processor [5], [15].

Performance monitoring counters 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 or the memory subsystem. The main difference between the models that use PMC lies in the types and amount of counters they employ. Bircher provides a valuable guideline for identifying the hardware events that should be selected as model inputs to estimate the power consumption of a processor [16]. For example, the author argues that events reflecting the fetched  $\mu$ -ops (micro-operations) per cycle minimize the model's estimation error while events reflecting  $\mu$ -ops retired may increase the estimation error. Isci and Martonosi [6] express the power consumption of a Pentium 4 CPU as the sum of the power consumption of the processor's 22 major subunits. The idea is further refined by Bertran et al. [17] who determine more than 25 architectural components and classify them into three categories, namely, in-order engine, out-of-order engine, and memory. The authors remark that the activities of some of the components in the in-order engine cannot be detected with distinct PME. The memory engine is further divided in three parts: L1 cache, L2 cache, and the main memory, the latter including the front side bus. Likewise, the out-

of-order engine is divided in three parts: INT unit, FP unit, and SIMD unit. Accordingly, the authors identify eight so-called power components and develop 97 micro-benchmarks that should stress each of these components in isolation under different scenarios. Their aim is to detect those PME that reflect the activity level of these components best.

Chen et al. identify five hardware events showing strong correlation with the power consumption of a processor [18]. These are the number of L1 data cache references per second, number of L2 cache references per second, number of L2 cache misses per second, number of floating point instructions retired per second, and number of branch instructions retired per second, respectively. The active power consumption of the processor is then expressed as a linear and weighted combination of these event rates. Singh et al. first classify performance monitoring counters into four basic categories, namely, Float Point Units, Memory, Stalls, and Instructions Retired, then they identify one performance monitoring counter from each category based on the exhibition of a strong correlation with the power consumption of the processor [19]. For their case, the performance monitoring counters they identify, with respective order, are: L2 CACHE MISS:ALL, RETIRED  $\mu$ -ops, RETIRED MMX AND FP INSTRUCTIONS:ALL, and DISPATCH STALLS. Then they apply a piece-wise linear regression function to estimate the power consumption of a processor. Lewis et al. argue that the power consumption of a processor due to a specific workload directly correlates with the change in core die temperature and the ambient temperature per processor [20]. Therefore, they propose a superposition model in which the temperature information is used together with transaction events in high speed data buses and L2 cache-miss events.

Dhiman et al. employ a combination of PMC and CPU utilization to estimate the power consumption of a virtual machine [21]. The PMC consists of instructions per cycle (IPC), memory accesses per cycle (MPC), and cache transactions per cycle (CTPC). The input values form a vector  $x = (x_{ipc}, x_{mpc}, x_{ctpc}, x_{util}, x_{pwr})$  which are assigned to different classes (or bins). Depending on the CPU utilization, bin sizes range from 20 percent utilization (5 bins) to 100 percent CPU utilization (1 bin). Within every bin the data vectors are quantized using a Gaussian Mixture Vector Quantization (GMVQ) technique [22]. The outputs of this quantization are multiple Gaussian components  $g_i$  that are determined by the mean and covariance of  $m = (m_{ipc}, m_{mpc}, m_{ctpc}, m_{util}, m_{pwr})$ . To train the model, the  $x_{pwr}$  component is removed from the  $x$  vectors and the resulting vectors are compared with those in the training set. The GMVQ algorithm then finds the nearest vector  $m$  in the training set to the input vector, and the value for the  $m_{pwr}$  component is retrieved. The retrieved value is compared to the actual  $x_{pwr}$  value from which the accuracy of the model is determined. This part of the training phase is repeated multiple times with different sizes of utilization bins. The bin size resulting the smallest error is selected as the model parameter. The same method is applied during the running phase: PMC values are obtained and the vector is compared to the model where the GMVQ algorithm finds the nearest (training) vector and returns the average power value.

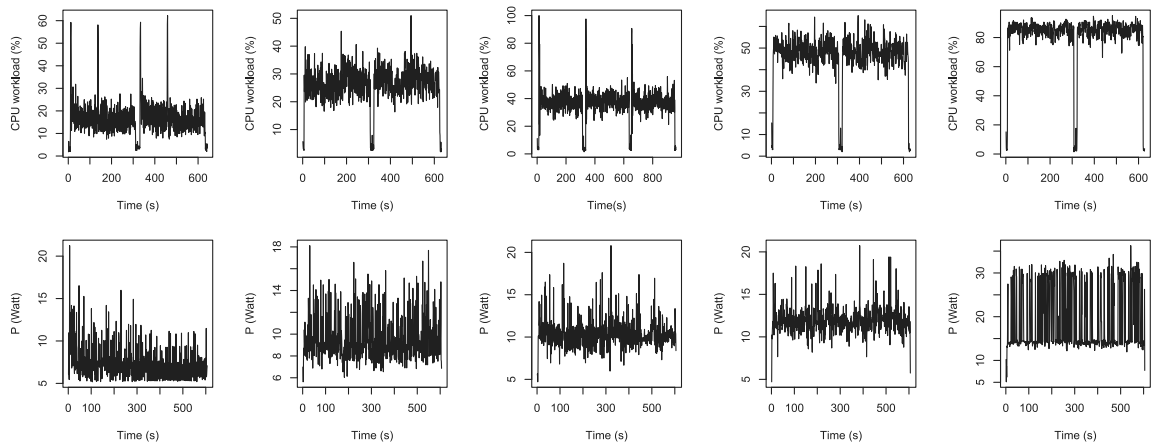


Fig. 1. The relationship between the CPU utilization and the power consumption of a D2461 Siemens-Fujitsu server running an online music search and download application. The request rate to the application was from left to right: 100, 200, 300, 400, and 800 requests per minutes.

There are some challenges with employing hardware performance counters: First, one should have knowledge of the low-level counters in order to be able to translate hardware events into a power consumption profile. Second, the identification of the relevant counters is dependent on the nature of the benchmark and the processor architecture. Third, in most processor architectures, one may be restricted by the number of counters that can be read at the same time. In contrast, our approach deals with a single quantity and makes use of its statistics which can be obtained in any platform. Furthermore, a stochastic model enables us to deal with variations in measurement resolutions as well as unpredictable and unobservable dynamics both inside the processor and the workload. Our approach is most useful for servers whose workload size exhibits considerable fluctuation. In which case the CPU utilization reveals sufficient statistics about the power consumed by the processor.

Our model employs the statistics of CPU utilization information only. This makes the model lightweight and the input easily accessible virtually in any sever platform. The usefulness of CPU utilization is first observed by Fan et al. [23] who report a remarkable estimation accuracy. Unfortunately, the authors do not provide a detail statistical account pertaining to the model's performance. A work closer to ours is the one proposed by Pedram and Hwang [24] in which the relationship between the power consumption and the utilization level of a multicore processor (a quad core Xeon E5410 processor) is investigated. The authors employ CPU-, memory-, and IO-intensive workloads; configure the processor to operate at different frequencies; and vary the number of active cores. Their investigation reveals that regardless of the type of workload and the processor's configuration, the power consumption and the utilization level are related linearly. However, each configuration results in different coefficients in the linear equation. Our investigation partially confirms this observation. The main difference between their approach and our approach is that while they consider the two quantities as deterministic quantities, we model them as random variables. The advantage of our approach is that the statistical properties (max, min, mean, CDF, variance, pdf, autocorrelation, etc.) of one of the quantities can

be sufficiently determined from the statistics of the other quantity. This aspect is useful for making energy-aware planning, as we already mentioned in Section 1.

### 3 MODEL INPUT

Through repeated experiments, we have observed that a strong correlation existed between the power consumption and the CPU utilization of a multicore processor. For example, Fig. 1 shows the CPU utilization and the power consumption of a D2461 Siemens-Fujitsu server integrating a 4 GHz AMD Athlon 64 dual core processor running an online music search and download application. The application was randomly downloading music files (with an average size of approx. 4 MB) at 100, 200, 300, 400 and 800 requests per minutes. As the request rate increases, the CPU utilization as well as the power consumption of the processor increases. Likewise, Fig. 2 shows the relationship between the CPU utilization and the power consumption of a D2581 Siemens-Fujitsu server integrating a 3.4 GHz Intel E8500 dualcore processor running the SPEC power\_ss2008 benchmark, which utilizes the entire utilization spectrum.

Arguably a processor may consume a different amount of power when executing different types of tasks even though it may have the same CPU utilization statistics for these tasks. For example, a 3.6 GHz Intel I5-680 dualcore processor consumes on average 37.98 W ( $p_{io}$ ) when

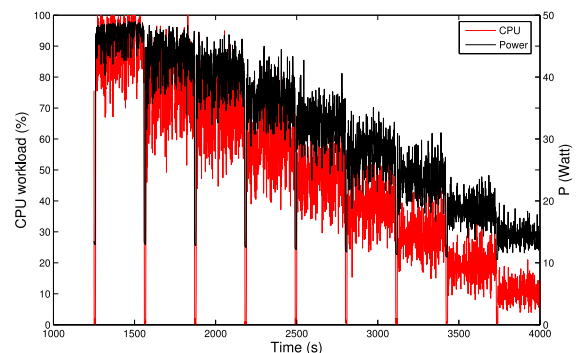


Fig. 2. The relationship between the CPU utilization and the power consumption of a D2581 Siemens-Fujitsu server running the SPEC Power (2008) benchmark.

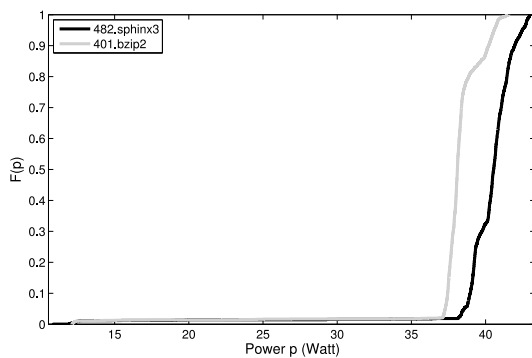


Fig. 3. The CDF of the power consumption of a 3.6 GHz Intel I5-680 dualcore processor executing the 482.sphinx3 (float point operation) and 401.bzip (integer operation) from the SPEC CPU 2006 benchmarks at approx. Sixty percent utilization level.

executing the 401.bzip2 benchmark (an integer operation) at approx. Sixty percent CPU utilization and 40.08 W ( $p_{fpo}$ ) when executing the 482.sphinx3 benchmark (a floating point operation)<sup>1</sup> at approx. Sixty percent CPU utilization. Furthermore, the processor consumes on average 8 W when it is idle ( $p_i$ ). In this case, ( $\Delta_{active} = |p_{fpo} - p_{io}|$ ) is significantly smaller than either ( $\Delta_{io,idle} = p_{io} - p_i$ ) or ( $\Delta_{fpo,idle} = p_{fpo} - p_i$ ). If the processor experiences a frequent idle state, which is not unusual in real world servers, then  $\Delta_{io,idle}$  and  $\Delta_{fpo,idle}$  dominate  $\Delta_{active}$ .

Fig. 3 shows the cumulative distribution functions of the power consumption of the Intel I5-680 dualcore processor when executing the two benchmarks. Whereas the distribution functions show a slight variation in power consumption for a comparable CPU utilization level, the variation is remarkably small. If the processor executes similar tasks, then  $\Delta_{active}$  becomes even smaller.

We shall show in this paper that our model accurately captures the statistics of the power consumption of a processor from the statistics of the CPU workload, provided that the CPU workload experiences significant fluctuations, including the existence of an idle state power consumption. We have tested our model with two different server platforms and three different benchmarks and the results we observed are encouraging.

## 4 EXPERIMENTAL SETTING

The power consumption of a processor has “static” and dynamic components. The “static” power consumption is required to power the processor and to make it ready to do some work. This component does not depend on the workload of the processor. It is not really static but it can be considered as such because its variation in time is small when compared to the variation of the dynamic component. The dynamic component, on the other hand, depends on the workload of the processor. Moreover, it is the sum total of the dynamic power consumptions of the different architectural components of the processor.

A piece of workload is a high-level request to the processor and consists of several low-level tasks such integer operations, floating point operations, read and write operations to the memory or the disk, etc. In general, the workload of a

processor of an Internet server should be modeled as a stochastic process, since it is difficult to determine in advance the type and the size of the workload that will arrive at the processor. We assert that there is a strong correlation between the statistics of the processor’s workload and the statistics of the processor’s utilization. Therefore, examining the statistics of the processor’s utilization enables us to estimate the power consumption of the processor due to the workload. Because of this assertion we use the term *workload* and CPU *utilization* interchangeably.

We used four types of server platforms to obtain statistics pertaining to the workloads and the power consumptions of single-core and dualcore processors. We tested our model on two of these platforms. The first one was built on a D2581 Siemens-Fujitsu motherboard integrating a 3.4 GHz Intel E8500 dualcore processor. The second server was built on a D2461 Siemens-Fujitsu motherboard integrating a 4 GHz AMD Athlon 64 dual core processor. Our model was able to estimate the power consumption of the two servers with comparable accuracy. We will frequently use the statistics of the D2581 (E8500) server to introduce our model.

The motherboard of the D2581 server provides two DC power connectors to supply the various subsystems with power. One of them is a 12 V, four-pole connector whereas the other is a 24-pole connector with 12, 5 and 3.3 V rails (among others). The 12 V rail of the four-pole connector is exclusively used by the voltage regulator of the processor—a two-phase voltage regulator controlled by an ISL 6326 Pulse Width Modulator (PWM) controller—to generate the processor core voltage. The output of each phase is supplied to the processor 50 percent of the time. This voltage regulator draws some amount of power from the 5 V rail of the 24-pole connector to control the PMW controller, but the amount is small. The 3.3 V is predominantly used by the Low Pin Count (LPC) IO controllers.

The power drawn through the 12 V and the 3.3 V rails of the 24-pole do not exhibit appreciable variations throughout the operation of the server, regardless of the type of workload we run on the server. This is also true for the other servers we employed. On the contrary, the powers drawn through the 12 V rail of the four-pole connector and the 5 V rail of the 24-pole connector exhibit variations that are proportional to the size of a workload. Of these two, however, the power drawn through the 12 V rail was much more sensitive to the variations in workload size; and its magnitude is significantly greater than the power drawn through the 5 V rail. We will consider the power drawn through the 12 V rail as the power consumption of the processor.

The measurement devices we employed were the Yokogawa WT210 digital power analyzers. The devices can measure DC as well as AC power consumption at a maximum rate of 10 Hz and a DC current between 15 A and 26 A with an accuracy of 0.1 percent.

## 5 STOCHASTIC MODEL

Before we begin with the introduction of our model, we will 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 a probability

1. <http://www.spec.org/cpu2006/>.



distribution function (or simply a distribution function) while a lower case  $f$  refers to a probability density function (or simply a density function).

The central question we would like to address in this article can be stated as follows: Suppose we provide a processor a piece of workload and it executes the workload within a certain amount of time. If we study the distribution or the density functions of the CPU utilization and the power consumption of the processor during this period, is it possible to estimate a quantitative relationship between the CPU utilization and the power consumption of the processor? We will show that by modeling the processor as a non-linear memoryless system with a stochastic input we can answer this question.

Thus, the CPU utilization and the dynamic power consumption of a processor will be characterized as random variables  $\mathbf{w}$  and  $\mathbf{p}$ , respectively, and our aim will be to express the statistics of  $\mathbf{p}$  in terms of the statistics of  $\mathbf{w}$  and vice versa. In Section 6, we will experimentally demonstrate that the assumption of a memoryless system is plausible.

### 5.1 Known Relationship

To highlight our point, we shall begin by assuming that the relationship between  $\mathbf{w}$  and  $\mathbf{p}$  is already known. Hence, we wish to determine the distribution of one of the random variables (the one whose statistics we do not know) in terms of the other (whose statistics we do know). For example, if the power consumption of a processor is expressed as

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

then,  $F(p) = P\{\mathbf{p} \leq p\} = P\{a\mathbf{w} + b \leq p\} = P\{\mathbf{w} \leq \lceil \frac{p-b}{a} \rceil\} = F_w(\lceil \frac{p-b}{a} \rceil)$ , 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  $\frac{d}{dp}(F(p)) = \frac{1}{a} f_w(\lceil \frac{p-b}{a} \rceil)$ , where  $f_w(p)$  refers to the density of  $\mathbf{w}$  expressed in terms of  $p$ . If, for instance, the density of  $\mathbf{w}$  is exponential:  $f(w) = \lambda e^{-\lambda w} U(w)$ ,  $\lambda > 0$ , then,  $f(p) = \frac{\lambda}{a} e^{-\lambda(\frac{p-b}{a})} U(\frac{p-b}{a})$ .

One may ask why this knowledge is important. The answer is straightforward. For example, if we have  $F(w)$ , we will be able to determine from it the probability that the processor power consumption exceeds a certain threshold and how long it may stay above or below this threshold. This can be done even before the workload is executed, so that a scheduler can plan when and where to execute it.

### 5.2 Unknown Relationship

A more interesting problem is determining the relationship between the two random variables given that we have their distribution and density functions. Alternatively, we can state the problem as follows: we are given the distribution of  $\mathbf{w}$  and wish to find a function  $g(\mathbf{w})$  such that the distribution of  $\mathbf{p} = g(\mathbf{w})$  equals a specified function  $F(p)$ . This is typically the case when we attempt to estimate the power consumption of a processor's workload. Initially, the processor can be given a workload of known statistics and the power consumption of the workload can be measured and analyzed. Then, we can establish the relationship between the two random variables. Once the relationship is established, the power consumption of the processor can be estimated for any arbitrary workload.

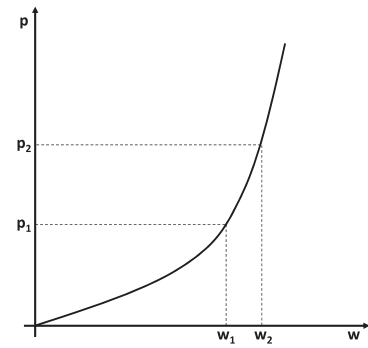


Fig. 4. Exploiting the one-to-one relationship between  $\mathbf{p}$  and  $\mathbf{w}$  and the monotonic nature of distribution functions to determine a quantitative relationship between  $\mathbf{w}$  and  $\mathbf{p}$ .

If the relationship between  $\mathbf{p}$  and  $\mathbf{w}$  can be considered as a one-to-one function, i.e., every element of the range of  $\mathbf{p}$  corresponds to exactly one element of the domain of  $\mathbf{w}$ ,<sup>2</sup> then  $P\{\mathbf{p} \leq p_i\}$  equals to  $P\{\mathbf{w} \leq w_i\}$  because  $\mathbf{p} \leq p_i$  if and only if  $\mathbf{w} \leq w_i$ . This can be better visualized in Fig. 4 which displays a one-to-one function. From the figure it is apparent that the value  $p_2$  corresponds to  $w_2$ . Therefore,  $P\{\mathbf{p} \leq p_2\}$  corresponds to  $P\{\mathbf{w} \leq w_2\}$ . Similarly,  $P\{\mathbf{p} \leq p_1\}$  corresponds to  $P\{\mathbf{w} \leq w_1\}$ . From this, we can conclude that for a one-to-one function:

$$F(p_i) = F(w_i), \quad \forall i \in \mathbb{R}. \quad (2)$$

Subsequently, using Equation (2), we can express  $\mathbf{p}$  in terms of  $F(p)$  and  $F(w)$  as follows<sup>3</sup>:

$$p = F_p^{-1}(F_w(w)), \quad (3)$$

where  $F_p^{-1}$  refers to the inverse of  $F(p)$  [25]. For example, if we observe that the distribution of the power consumption of a processor for an exponentially distributed workload  $(1 - e^{-\lambda w})$  is uniformly distributed in the interval (10, 50), then, using Equation (2):  $(1 - e^{-\lambda w}) = \frac{1}{40}p$ , from which,  $p = 40(1 - e^{-\lambda w}) = 40F(w)$ .

## 6 EXPERIMENT

In this section, we illustrate how we applied the concept we developed in Section 5 to establish the relationship between  $\mathbf{w}$  and  $\mathbf{p}$  of a single-core processor. To make the discussion tractable, we will give a summary of what will follow shortly.

In the first step, we disabled one of the cores of the E8500 Intel processor. Then, we generated a one hour uniformly distributed CPU-bound workload and supplied it to the server. The workload is produced by a program that computes the convolution of discrete functions (a mixture of integer and floating point operations). The CPU is utilized 100 percent when the program is executed, but remains idle when the program is not executed. In order to generate

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

3. From the relationship between variables and functions, we know that if  $y = f(x)$ , then  $x = f^{-1}(y)$ . If  $y$  can be expressed in the form of another function  $g(w)$ , then  $x = f^{-1}(g(w))$ .

TABLE 1  
Summary of the Estimated Parameters of  $F_p(p)$   
( $p_1p^2 + p_2p + p_3$ ) for the Intel E8500 Dual Core Processor

Density of $\mathbf{w}$		U(0,100)	U(10,90)	U(20,80)	U(30,70)
Range of $\mathbf{p}$ (in Watt)	$p_{low}$	9.45	10.25	10.34	10.24
	$p_{high}$	32.15	32.48	31.32	30.81
Parameter's value	$a_1$	-0.001026	-0.001026	-0.001263	-0.000769
	$a_2$	0.07765	0.07765	0.09715	0.07964
	$a_3$	-0.6136	-0.6136	-0.9053	-0.7661

Minimum confidence bounds of 95 percent.

the desired density function of the workload, we divided time into a set 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. The function generates uniformly distributed random numbers. For each time window, we picked out one of the uniformly distributed random numbers and determined with it the portion of the one-second time slot the CPU should be fully utilized by the convolution operation. In order to avoid unpredictable performance, 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. We measured the overall power consumption of the server as well as the power drawn through the 12 V rail of the four-pole connector.

In the second step, we approximated the probability distribution function of the actual power consumption using R's `nls` curve fitting tool. This step is useful to obtain  $g(\mathbf{w})$  using  $p = F_p^{-1}(F(w))$ .

Once we have  $g(\mathbf{w})$ , we tested the model's accuracy by using both a custom-made workload and standard benchmarks (SPEC power\_ssj2008 benchmark and the Apache benchmarking tool for an HTTP server<sup>4</sup>). The custom-made workload was the same discrete convolution, but this time it had an exponential and normal (for the AMD processor) distributions instead of a uniform distribution. The SPEC power\_ssj2008 "is the first industry-standard SPEC benchmark that evaluates the power and performance characteristics of volume server class and multi-node class computers".<sup>5</sup> The full SPEC power benchmark run for 70 minutes. The Apache benchmarking tool is used to evaluate the number of requests the Apache installation can serve per second. We used it to download video files by varying the request arrival rate from 500 to 5,000 requests per minute. The size of the videos we downloaded varied between 5 and 100 MB. The custom-made benchmark was tested on both the E8500 and the AMD servers, but the standard benchmarks were tested on the E8500 server only. Finally, we compared  $f(p)$  and  $F(p)$  of the actual power consumed during the test phase with  $f(p)$  and  $F(p)$  of the power we estimated using  $p = F_p^{-1}(F(w))$ .

## 6.1 Relationship between $w$ and $p$

To establish the relationship between  $\mathbf{w}$  and  $\mathbf{p}$ , we generated five uniformly distributed workloads using the discrete convolution operation, i.e., the CPU workload in percent: U

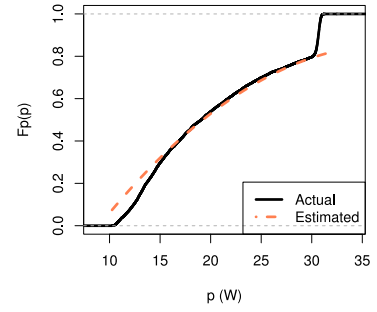


Fig. 5. The actual (measured) and estimated  $F(p)$  for the Intel E8500 dual core processor for  $U(0, 100)$ .

(0, 100), U(10, 90), U(20, 80), U(30, 70), and U(40, 60). There was, however, a certain disparity between the intended (theoretical) distribution and the distribution of the actual workload we generated. For each test case, we measured the power consumption of the processor and plotted its  $F(p)$ . Then, using the `nls` curve fitting toolbox, we approximated  $F(p)$ . Except for U(40, 60),  $F(p)$  can be best approximated by a quadratic function,  $F(p) = a_1p^2 + a_2p + a_3$ , where  $a_1$ ,  $a_2$ , and  $a_3$  are the coefficients of the quadratic function. For U(40, 60),  $F(p)$  is best approximated by the linear function  $a_1p + a_2$ , where  $a_1 = 0.05075$  and  $a_2 = 0.5163$  in the interval  $[10.17, 30.13]$ . Table 1 summarizes the parameters of the distribution functions that are approximated by the quadratic functions. Fig. 5 displays the experimental and approximated  $F(p)$  for U(0, 100) workload.

Hence, for the quadratic functions, we have

$$F(p) = a_1p^2 + a_2p + a_3 \quad p_{low} \leq p \leq p_{high}, \quad (4)$$

$$F(w) = \frac{w - b}{c - b} \quad b \leq w \leq c, \quad b, c > 0, \quad (5)$$

where  $b$  and  $c$  are the lower and upper bounds of the uniformly distributed workload. From the experiment results in Table 1, it is clear that  $a_1 < 0$ ,  $a_2 > 0$ , and  $a_3 < 0$ . With this knowledge and inserting Equations (4) and (5) into Equation (3), we obtain

$$\mathbf{p} = \frac{-a_2 + \sqrt{a_2^2 - 4a_1 \times (a_3 - (\mathbf{w} - b)/(c - b))}}{2 \times a_1}. \quad (6)$$

Equation (6) can be expressed as

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

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

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 determine the distribution and density of  $\mathbf{p}$  in terms of the distribution and density of  $\mathbf{w}$  for any arbitrary  $\mathbf{w}$ . Earlier, we showed that  $F(p)$  can be expressed as  $P\{\mathbf{p} \leq p\} = P\{g(\mathbf{w}) \leq p\}$ . Hence,

$$F(p) = P\left\{\left(K_1 + (K_2\mathbf{w} + K_3)^{\frac{1}{2}}\right) \leq p\right\} \quad b \leq \mathbf{w} \leq c. \quad (8)$$

4. <http://httpd.apache.org/docs/2.2/programs/ab.html>.

5. [http://www.spec.org/power\\_ssj2008/](http://www.spec.org/power_ssj2008/).

Expressing Equation (8) in terms of  $F(w)$  yields

$$F_w(p) = P\left\{ \mathbf{w} \leq \frac{(p - K_1)^2 - K_3}{K_2} \right\} \quad b \leq \mathbf{w} \leq c, \quad (9)$$

which is the same as  $F_w\left(\frac{(p - K_1)^2 - K_3}{K_2}\right)$ . Likewise, the density of  $\mathbf{p}$  can be expressed as

$$f(p) = \left| \frac{2}{K_2} (p - K_1) \right| f_w\left(\frac{(p - K_1)^2 - K_3}{K_2}\right). \quad (10)$$

## 6.2 Theoretical $F(p)$ and $f(p)$

Using the relationship expressed in Equation (7) and the parameters obtained from the experiment and listed in Table 1, 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 and normally distributed workloads. In the section that follows we shall compare the theoretical results with the ones we obtained from experiments.

### 6.2.1 Exponentially Distributed Workload

When  $\mathbf{w}$  is exponentially distributed ( $f(w) = \lambda e^{-\lambda w}$ ;  $\mu = \frac{1}{\lambda}$ ,  $w > 0$ ), its distribution function equals

$$F(w) = 1 - e^{-\frac{w}{\mu}}. \quad (11)$$

And the probability density function of  $\mathbf{p}$  can be expressed as follows:

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

The distribution function of  $\mathbf{p}$  is expressed as

$$F(p) = F_w(p) = 1 - e^{-\left(\frac{(p - K_1)^2 - K_3}{K_2}\right)/\mu}, \quad (13)$$

where  $p$  is in the interval  $[p_{low}, p_{high}]$  (see Table 1) and  $F_w(p)$  refers to the probability distribution function of  $\mathbf{w}$  expressed in terms of  $\mathbf{p}$ .

### 6.2.2 Normally Distributed Workload

Similarly, when  $\mathbf{w}$  is normally distributed, its distribution function is given as

$$F(w) = G\left(\frac{w - \mu}{\sigma}\right) = \frac{1}{2} \left( 1 + \operatorname{erf}\left(\frac{w - \mu}{\sqrt{2}\sigma}\right) \right), \quad (14)$$

where  $\operatorname{erf}(w) = \frac{2}{\sqrt{\pi}} \int_0^w e^{-t^2} dt$ . Therefore, the distribution and the density functions of  $\mathbf{p}$ , using the relationship expressed in Equation (7), can be expressed using Equations (15) and (16), respectively,

$$F_w(p) = \frac{1}{2} \left( 1 + \operatorname{erf}\left(\frac{\left(\frac{(p - K_1)^2 - K_3}{K_2}\right) - \mu}{\sqrt{2}\sigma}\right) \right), \quad (15)$$

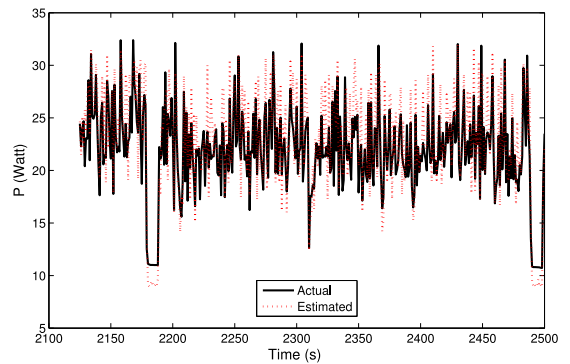


Fig. 6. A snapshot of the actual and estimated power consumption of the Intel E8500 single-core processor when executing the SPEC power benchmark.

$$f(p) = \left| \frac{2}{K_2} (p - K_1) \right| \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(\left(\frac{(p - K_1)^2 - K_3}{K_2}\right) - \mu\right)^2}{2\sigma^2}} \quad (16)$$

In Equation (15) and (16),  $p$  is in the interval  $[p_{low}, p_{high}]$  (see Table 1).

## 6.3 Experimental $F(p)$

After having established the relationship between  $\mathbf{p}$  and  $\mathbf{w}$ , we tested the validity of our model by generating normally distributed and exponentially distributed workloads and by executing the workloads on the two servers. The workloads with the exponential distribution had  $\mu = 5, 10, 15$  and  $20$ . The workload with the normal distribution (for the AMD processor) had the following parameters:  $N(\mu = 30, \sigma = 5)$ . We also employed two additional standard benchmarks, namely, the SPEC power benchmark and the Apache benchmarking tool to test our model. These benchmarks stressed the CPU in quite different ways.

Equation (7) can be directly applied to estimate the instantaneous relationship between  $\mathbf{w}$  and  $\mathbf{p}$  as well as the statistics of  $\mathbf{p}$  because of Equation (9). Fig. 6 displays the actual and estimated instantaneous power consumption of the E8500 single-core processor when it executed the SPEC power benchmark. As can be seen from the figure, the model estimated the power consumption fairly accurately and followed the variation pattern accurately as well.

To estimate the theoretical distribution functions of all the test cases, we used the coefficients of  $F(p)$  obtained for the  $U(0, 100)$  in Table 1, since the domain of the uniformly distributed workload in the interval  $(0, 100)$  subsumes the domains of all the test cases.

Fig. 7 (left) displays the theoretically estimated and the experimentally obtained  $F(p)$  for the exponentially distributed workloads of the E8500 processor. The same figure (right) displays the theoretically estimated and the experimentally obtained  $F(p)$  for the exponentially (left) and normally (right) distributed workloads for the AMD Athlon 64 processor. As can be seen from both figures, when the test workload was similar in type with the training workload (for our case, the convolution operation), its power consumption could be accurately predicted regardless of the processor type (with an average error of 0.76 percent) even though the statistics of the workloads were dissimilar.

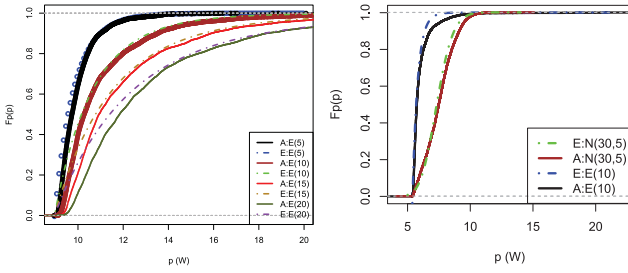


Fig. 7. The actual (A) and estimated (E)  $F(p)$  of the Intel E8500 (left) and AMD (right) processors when executing exponentially distributed workloads.

Our model estimated the power consumption of the Apache workload with an estimation error of 2.7 percent (see Fig. 9 (extreme left)). The Apache server operated in one of the two extreme utilization regions: in the region near 0 percent when the request rate was small or in the region near 100 percent when the request rate was high, as shown in Fig. 8 (left).

The model's estimation error increased to 7.37 percent on average for the SPEC power benchmark. The SPEC power benchmark utilized the CPU in a variety of manners, with different types of operations and utilizing the various architectural components with different intensities. Fig. 8 (right) shows the range of the density function of the CPU workload when executing the SPEC power benchmark. The theoretically estimated and experimentally obtained  $F(p)$  for the SPEC power benchmark is displayed in Fig. 9 (the second from left).

## 7 MULTICORE MODEL

In a multicore processor, it is possible to capture (and condition) the distribution of the workloads of individual cores but it is difficult to separately measure the power consumption of individual cores. Therefore, the best approach to represent the relationship between the workload and the power consumption of the processor is to use a Multiple-Input-Single-Output (MISO) memoryless stochastic model. In this case, we should first determine the overall workload of the processor,  $w_t$ , and then establish a relationship between  $w_t$  and  $p$ .

### 7.1 Workload Model

The overall workload of a multicore processor is given as  $w_t = w_1 + w_2 + \dots + w_n$ , where  $w_i$  refers to the workload of the  $i$ th core of the processor. The workloads of the individual cores are random variables and the summation rule of random variables applies to determine the density and the distribution functions of  $w_t$ .<sup>6</sup> For a dualcore processor,

6. The density of  $w_t$  tends to be normally distributed as the number of cores becomes large, complying with the Central Limit Theorem. The power consumption estimation model for a multicore processor, in addition to the workload of individual cores, should, therefore, take the power consumption due to the inter-core communication into account. This task requires knowledge of the processor's architecture. The extra power consumption can be neglected if we assume that  $w_1, w_2, \dots, w_n$  are statistically independent, which is the case for our model. With this assumption, as far as establishing the relationship between  $p$  and  $w_t$  is concerned, there is no difference between the models of a multicore and a dualcore processor. Therefore, we will focus on a dualcore processor to make the investigation tractable.

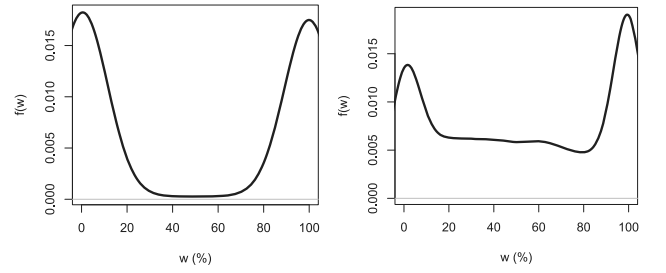


Fig. 8. The density of the Intel E8500 processor workload when executing the Apache benchmark (left) and the SPEC power benchmark (right) in a single-core mode.

given the random variables  $w_1$  and  $w_2$  and  $w_t = w_1 + w_2$ , the distribution of  $w_t$  can be expressed as

$$\begin{aligned} F(w_t) &= P\{w_t \leq w_t\} = P\{(w_1 + w_2) \leq w_t\} \\ &= P\{(w_1 + w_2) \in D_{w_t}\}, \end{aligned} \quad (17)$$

where  $D_{w_t}$  in the  $w_1 w_2$  plane represents the region in which the inequality  $(w_1 + w_2) \leq w_t$  is satisfied. The density function of  $w_t$  can be obtained as the convolution of  $f_{w_1}(w_1)$  and  $f_{w_2}(w_2)$ <sup>7</sup>:

$$f_{w_t}(w_t) = \int_0^{w_t} f_{w_1}(w_t - w_2) f_{w_2}(w_2) dw_2 \quad w_t \geq 0, \quad (18)$$

where  $f_{w_1}(w_t - w_2)$  is  $f_{w_1}(w_1)$  expressed in terms of  $w_1 = w_t - w_2$ .

### 7.2 Relationship between $w_t$ and $p$

In the single-core processor model, we used a uniformly distributed workload to establish the relationship between  $p$  and  $w$ . The uniform distribution was chosen because it simplified the calculation of  $p = F_p^{-1}(F(w))$ . For a dualcore processor, however, conditioning the distribution of the overall workload  $w_t$  to assume a uniform distribution was difficult, because one has to deal with multiple monotonically increasing distribution functions. The suitable approach is to generate uniformly distributed  $w_1$  and  $w_2$  in the interval  $[0, 100$  percent] and to supply them to the two cores. As long as the operating system (scheduler) controls the workload distribution between the two cores, we can assume that  $w_1$  and  $w_2$  are statistically independent. Consequently, the distribution of  $w_t$  (using Equation (17)) is given as<sup>8</sup>:

$$F(w_t) = \begin{cases} (0.5 \times 10^{-4}) w_t^2, & 0 \leq w_t \leq 100, \\ 1 - (0.5 \times 10^{-4}) (200 - w_t)^2, & 100 < w_t \leq 200. \end{cases} \quad (19)$$

7. Sometimes we add subscripts to density and distribution functions in order to avoid confusion. For example,  $f_{w_1}(w_1)$  should be understood as the density function of  $w_1$ .

8. Since the overall workload is a function of two independent, uniformly distributed random variables, its distribution is determined as:

$$F(w_t) = \begin{cases} \int_0^{w_t} \int_0^{w_t - w_2} dw_1 dw_2, & 0 \leq w_t < 100, \\ 1 - \int_{w_t - 100}^{100} \int_{w_t - w_2}^{100} dw_1 dw_2, & 100 \leq w_t < 200. \end{cases} \quad (20)$$



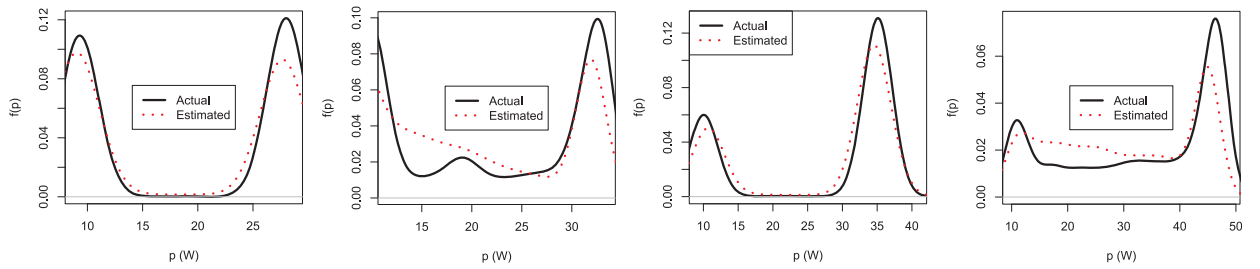


Fig. 9. The actual and estimated  $f(p)$  of the power consumption of the Intel E8500 processor when executing, from left to right, the Apache benchmark (single-core mode), the SPEC power benchmark (single-core mode) the Apache benchmark (dualcore mode), and the SPEC power benchmark (dualcore mode).

Likewise, the density of  $w_t$  (after differentiating Equation (19)) is given as:

$$f(w_t) = \begin{cases} (10^{-4})w_t, & 0 \leq w_t \leq 100, \\ (10^{-4})(200 - w_t), & 100 < w_t \leq 200. \end{cases} \quad (21)$$

Fig. 10 (left) shows the actual and the theoretical distributions of the overall workload (the discrete convolution operation we mentioned in Section 6) we executed for one hour on the E5800 dualcore processor (each core executing a uniformly distributed workload simultaneously).

A closer examination of both the theoretical and the experimental  $F(w_t)$  reveals that it has a convex property for  $0 \leq w_t \leq 100$  and a concave property for  $100 < w_t \leq 200$ . Through repeated test we have observed that this property remains unchanged, confirming that  $w_1$  and  $w_2$  can indeed be considered statistically independent and estimating the actual distribution with Equation (20) is reasonable. Fig. 10 (right) displays the distribution of the processor's corresponding power consumption (the solid line). Similar to  $F(w_t)$ ,  $F(p)$  has two components, namely, for  $0 \leq w_t \leq 30.5$  it exhibits a convex property while for  $30.5 < w_t \leq 50$ , it exhibits a concave property. We used Matlab to approximate the actual distribution with the following expression:

$$F(p) = \begin{cases} (2.47 \times 10^{-4})(w_t - 10.5)^2, & 10.5 \leq p \leq 30.5, \\ 1 - (2.37 \times 10^{-4})(42.2 - (w_t - 10.5))^2, & 30.5 < p \leq 50. \end{cases} \quad (22)$$

Clearly, associating the convex part of  $F(p)$  with the convex part of  $F(w_t)$  and the concave part of  $F(p)$  with the concave part of  $F(w_t)$  shows that there is a linear relationship between  $w_t$  and  $p$ :

$$p = K_4 w_t + K_5 \quad K_4 > 0. \quad (23)$$

With,

$$F(p) = P\{K_4 w_t + K_5 \leq p\} = F_{w_t}\left(\frac{p - K_5}{K_4}\right). \quad (24)$$

Combining Equation (19) and Equation (24) yields:

$$F(p) = \begin{cases} \left(\frac{0.5 \times 10^{-4}}{K_4^2}\right)(p - K_5)^2, & 10.5 \leq p < 30.5, \\ 1 - \left(\frac{0.5 \times 10^{-4}}{K_4^2}\right)(200K_4 - p - K_5)^2, & 30.5 \leq p \leq 50, \end{cases} \quad (25)$$

where  $K_4 = 0.2$  and  $K_5 = 10.5$ .

### 7.3 Theoretical $F(p)$ and $f(p)$

If  $w_1$  and  $w_2$  are exponentially distributed with the same  $\lambda_1 = \lambda_2 = \frac{1}{\mu}$ , the distribution of the overall workload  $w_t$ ,

$$\begin{aligned} F(w_t) &= P\{w_1 + w_2 \leq w_t\} \\ &= \begin{cases} 1 - \left(1 + \frac{w_t}{\mu}\right)e^{-\frac{w_t}{\mu}}, & 0 \leq w_t \leq 100, \\ 1 - \left(\frac{200 - w_t}{\mu} - 1\right)e^{-\frac{w_t}{\mu}} - e^{-\frac{200}{\mu}}, & 100 \leq w_t \leq 200. \end{cases} \end{aligned} \quad (26)$$

Therefore, the theoretical distribution of  $p$ , given the linear relationship between  $w_t$  and  $p$  is expressed as:

$$\begin{aligned} F(p) &= \begin{cases} 1 - \left(1 + \frac{p - K_5}{K_4 \mu}\right)e^{-\frac{p - K_5}{K_4 \mu}}, & 0 \leq w_t \leq 100, \\ 1 - \left(\frac{200K_4 - p + K_5}{K_4 \mu} - 1\right)e^{-\frac{p - K_5}{K_4 \mu}} - e^{-\frac{200}{\mu}}, & 100 \leq w_t \leq 200. \end{cases} \end{aligned} \quad (27)$$

### 7.4 Experimental $F(p)$

Similar to what we carried out in Section 6, we tested the validity of our model by using custom-made exponential workloads as well as the SPEC power benchmark and the Apache benchmarking tool. We carried out our test using the Intel E8500 dualcore processor. Fig. 11 (left) shows the exponentially distributed overall workload of the dualcore processor for  $\mu = 5$  and  $\mu = 15$  and in the same figure (right) the distribution of the actual and the estimated power consumptions of the dualcore processor while executing the exponentially distributed workloads is displayed. The linear relationship approximated better when  $\mu = 5$  (error = 0.3 percent) than when  $\mu = 15$  (error = 0.7 percent). The reason is that as the value of  $\mu$  increased, the CPU utilization fluctuation range increased as well, potentially increasing the deviation of the estimated power consumption from the actual power consumption. This problem becomes more visible when we consider the standard benchmarks.

The density of the workload generated by the Apache benchmarking tool was similar in pattern with the density of the single-core Apache workload. The density of the actual and estimated power consumption,  $f(p)$ , of the E8500 dualcore processor when executing the Apache benchmark is displayed in Fig. 9 (the third from left). The estimation error was comparable with that of the error for the single-core processor. During the execution of the SPEC power benchmark, the CPU utilization of the E8500 dualcore processor, similar to the case with the single core processor, occupied the entire CPU utilization spectrum with a visible

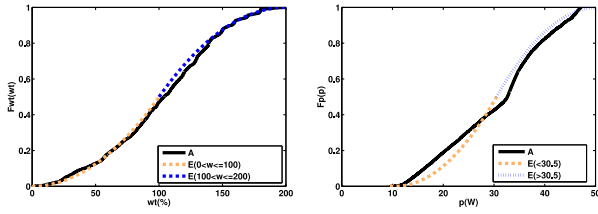


Fig. 10. (Left): The actual (A) and the estimated (E) distribution of the overall workload ( $F_{w_t}(w_t)$ ) of the E8500 dualcore processor as the convolution of two uniformly distributed workloads executing on each core. Right: The corresponding actual (A) and estimated (E)  $F(p)$ .

dominant utilization at the two ends of the spectrum (i.e., 0 and 100 percent). The density of the actual power consumption is shown in Fig. 9 (extreme right, the black solid line). There is an apparent deviation between the actual and the estimated  $f(p)$  in the interval (15, 33) Watt. The average estimation error amounts to 5.2 percent, a slight improvement compared to the single core model. It is noteworthy that in all the estimation assignments, both for the single-core and the dualcore models, our approach was responsive to the changes in the power consumption, which we consider as an indication of the quality of our model.

It must be remarked once again that our model is capable of estimating the instantaneous power consumption by directly applying Equation (23). Fig. 12 shows the actual and estimated instantaneous power consumption of the Intel E8500 dualcore processor when executing the SPEC power benchmark. 67 percent of the time, the model's estimation error was within  $\pm 5$  Watt range. The density of the estimation error for the instantaneous power is displayed in Fig. 13.

## 8 ERROR

There are three types of errors in our estimation model. The first type of error stems from the imperfection of the workload we used to train the model. The second type of error stems from the approximation of the actual (measured)  $F(w)$  and  $F(p)$  before applying Equations (7) and (23) (i.e., the error due to a curve-fitting process). The third type of error stems from the deviation of the estimated or theoretical  $F(p)$  (obtained by using the relationships expressed in Equations (7) and (23)) from the actual or measured  $F(p)$ . We refer to the last type of error as the estimation error.

The first and third types of error are inherent to all types of power consumption estimation errors. As long as the future workload of the processor is unknown and the server hosts a large number of services, it is difficult to train the model with a representative workload. For example, Bertran et al. [26] develop 97 different types of micro-benchmarks to deal with this problem. Specific to our model is the second type of error.

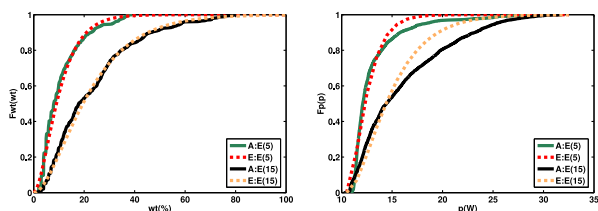


Fig. 11.  $F(w)$  of the exponentially distributed workloads executed on the E8500 dualcore processor(left) and the corresponding actual (A) and estimated (E)  $F(p)$  (right) –  $\mu = 5$  and  $\mu = 15$ .

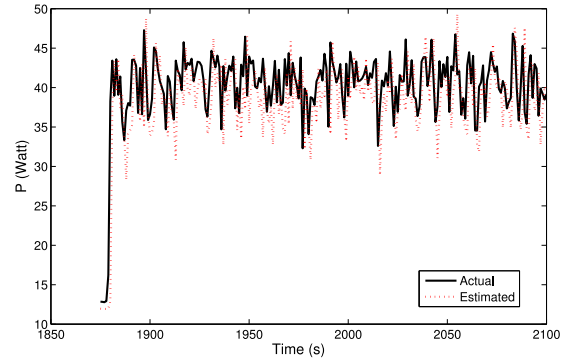


Fig. 12. A snapshot of the actual and estimated power consumption of the Intel E8500 dualcore processor while executing the SPEC power benchmark.

We use the root-mean-square error (RMSE) and the sum of squared errors (SSE) [27], [28] to quantify the model's errors. We use the RMSE to quantify the first two error types and the SSE to quantify the estimation error. These two quantities essentially measure the differences between the values estimated by our model and the values we actually observed (measured). The essential difference between them is that the former expresses the expected error value whereas the latter is an expression of the accumulated (total) error. We employ SSE to quantify the estimation error because the distribution (density) function describes the entire domain of a random variable.

Of the three types of errors, the one which affected our model most was the curve-fitting error. In the single-core processor, this can be seen in Fig. 5; the deviation between the actual and the approximated  $F(p)$  increased significantly for  $p > 30$  W. This region corresponds to  $w > 80\%$ . Hence, the estimation error of Equation (4) increased for  $w > 80\%$ . The magnitude of this error depends on how often the workload of the processor exceeds the 80 percent threshold, i.e.,  $\int_{80}^{100} f(p) dp$ . For example, for the SPEC power, this error amounts on average to 5.2 percent. For the Apache benchmarking tool it was 2 percent. In the dualcore processor, the error in the curve-fitting approximation lied in the coefficient  $K_4 = 0.2$  (see Equation (25)). The reason can be seen in Fig. 10; in the figure approximating the lower part of the actual power consumption (for  $w_t \leq 100$ ) resulted in  $SSE = 22.28$  and approximating the upper part ( $w_t > 100$ ) resulted in  $SSE = 1.556$ . The average RMSE of the curve-fitting process equals to 3.65 percent. Hence, when we used

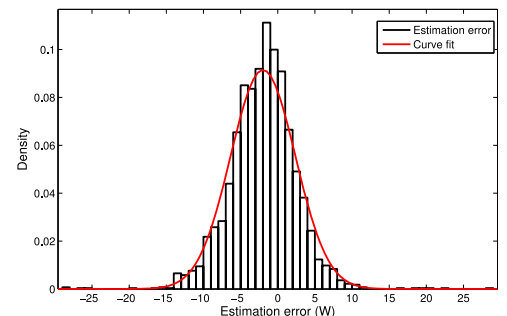


Fig. 13. The estimation error density for the instantaneous power consumption of the Intel E8500 dualcore processor when executing the SPEC power benchmark.

$K_4 = 0.2$ , the estimated power consumption was always less than the actual power consumption of the dualcore processor for all the test cases, irrespective of the type or the distribution of the workload. When we readjusted this value to  $K_4 = 0.35$ , the estimation error significantly reduced for all the test cases.

## 9 COMPARISON

The models which are closer in purpose as well as in approach (the creation of a lightweight model) to ours are the ones proposed by Fan et al. [23], Heath et al. [29], Economou et al. [30], and Bircher [16]. Fan et al. employ, similar to us, the CPU utilization as the input of a non-linear model together with an empirically obtained “correlation” or “calibration” factor. Similarly, Heath et al. employ CPU and disk utilization as the input parameters of a linear regression model. The model of Economou et al. employs events emitted from a selected number of performance monitoring counters in addition to CPU utilization, memory access, disk IO rate, and network IO rate. These three models have been comparatively evaluated by Rivoire in [14].

The evaluation was carried out on different server platforms, including a high-performance Xeon server (2x Intel Xeon E53459). On each system ran CPU-, memory- and IO-bound benchmarks: SPECfp, SPECint and SPECjbb (as CPU intensive workloads), stream (as a memory intensive workload), and ClamAV virus scanner, Nsort, and SPECweb (as IO intensive workloads). Overall the model of Economou et al. performed well, producing the smallest average estimation error. Even so, its performance on the Xeon system displayed stark, workload dependent variations. For example, it had an estimation error of 9.5 percent for the SPECint benchmark suite whereas the models of Fan et al. and Heath et al. scored an estimation error of less than 4.5 percent for the same benchmark. Moreover, the model of Heath et al. had an estimation error of 2.25 percent for the SPECfp benchmark while this was 8 percent for the model of Economou et al. This indicates that PMC-based models are not necessarily more accurate than models based on CPU utilization, but that accounting for nonlinear properties reduces the estimation error even when abstract model inputs are used. Having said this, the smallest maximum estimation error across all the benchmarks considered was observed with the model of Economou et al., whereas in the other models the difference between the smallest and the largest estimation error was considerably high, suggesting that these models can be unreliable for some benchmarks (workload types).

The model of Economou et al. was also evaluated by McCullough et al. in [31]. For best comparison, we consider here their results for the Intel Core i7-820QM10 and the SPECfp benchmarks (povray, soplex, namd, zeusmp, sphinx3). The average estimation error of the model was 4.23 percent which is comparable with the observation made in [14]. However, for other benchmarks, it performed poorly—up to 15 percent estimation error was observed. McCullough et al. explain that this is due to the existence of “hidden states” in the multicore CPU architecture producing nonlinear behavior that cannot be captured by the model.

The model of Birch et al. employs a single event captured by a performance monitoring counter—*fetch*

*micro-operations/cycle*—to establish a linear relationship between a CPU’s activity and its power consumption. The authors argue that the event representing the amount of fetched micro-operations better capture the CPU’s activity than the event representing the amount of micro-operations retired, since the number of fetched micro-operations comprises both retired and canceled micro-operations. They train and test their model with the SPEC2000 benchmark suite, splitting the benchmarks into 10 clusters. From each cluster one program is used for training the model while the remaining are used to test the model. In addition to the SPEC2000 suite, the authors employ custom-made benchmarks to examine the minimum (with minimum instruction per cycle) and the maximum (with maximum instruction per cycle) power consumption of the CPU. The model’s average estimation error was 2.6 percent. The authors further reduced this error to 2.5 percent by considering an additional event (*uop\_queue\_writes*) because floating point operations consist of complex microcode instructions that cannot be sufficiently captured by the previous event [32].

In terms of model complexity, our model is comparable with the models of Fan et al. and Bircher. The model of Economou et al. cannot be considered lightweight. Unlike the models of Fan et al. and Heath et al., our model was tested by considerably varying the CPU utilization. Furthermore, our test environment involves more diverse benchmarks, both standard and custom-made. In terms of estimation error, our model is comparable with the model of Economou et al. The maximum estimation error we observed was below 8 percent whereas this figure was below 10 percent for the model of Economou et al. However, the model of Economou et al. has been tested by a third party on a variety of server platforms. In contrast, our model was tested on two server platforms only. It will be interesting to see a third party evaluating our model and compare it with other contending models.

## 10 CONCLUSION

We characterized the power consumption of a processor and its workload as random variables and employed their probability distribution functions to establish a quantitative relationship between them:  $p = F_p^{-1}(F(w))$ . For a single-core processor, the relationship between  $\mathbf{w}$  and  $\mathbf{p}$  was best expressed as a quadratic relation whereas for a dual-core processor, the relationship was best expressed by a linear function.

Our approach is useful as long as the processor can be considered as a memoryless system with a stochastic input. A sufficient precondition for this assumption is that the autocorrelation of  $\mathbf{p}$ ,  $R_{pp}(t_2, t_1) \approx 0$  for  $t_2 \neq t_1$  if the autocorrelation of  $\mathbf{w}$ ,  $R_{ww}(t_2, t_1) \approx 0$  for  $t_2 \neq t_1$ . This requirement can be satisfied by carefully choosing a suitable sampling interval during measuring the power consumption of the server. We have experimentally observed that for a sampling interval of a few hundred milliseconds, this requirement can be satisfied.

We tested our models with custom-made workload as well as with standard benchmarks, namely, with SPEC Power ssj2008 benchmark and the Apache benchmarking tool. Unlike the SPEC CPU benchmark families or similar



benchmarks, the SPEC power benchmark stresses a processor with different magnitudes, generating a wider utilization spectrum, (0, 100 percent). This type of benchmark better represents the workload of servers hosting a large number of Internet applications. Likewise, the Apache benchmarking tool is specifically designed to test the performance of an Apache Internet installation.

Even though we trained the models with a predominantly integer and floating point operations, the custom-made program performed well for all the training cases. Understandably, the estimation error was larger for the SPEC power benchmark than for all the other test workloads. This is due to the wider workload spectrum. Of all the error types, the error due to the approximation of the actual  $F(w)$  and  $F(p)$  with a curve-fitting tool was the largest. Reducing this error is possible but it comes at the expense of making the models more complex. Even so, both models were able to detect changing characteristics in the power consumption of the processor. This aspect is desirable because it helps the server to quickly adapt to a changing workload.

## ACKNOWLEDGMENTS

This work has been partially funded by the German Research Foundation (DFG) under agreement SFB 912/1 2011.

## REFERENCES

- [1] R. Kumar, D. M. Tullsen, P. Ranganathan, N. P. Jouppi, and K. I. Farkas, "Single-ISA heterogeneous multi-core architectures for multithreaded workload performance," *SIGARCH Comput. Archit. News*, vol. 32, no. 2, pp. 64–75, Mar. 2004.
- [2] T. Horvath and K. Skadron, "Multi-mode energy management for multi-tier server clusters," in *Proc. 17th Int. Conf. Parallel Arch. Compilation Techn.*, 2008, pp. 270–279.
- [3] A. K. Coskun, T. S. Rosing, and K. Whisnant, "Temperature aware task scheduling in MPSOCS," in *Proc. Conf. Des., Autom. Test in Eur.*, San Jose, CA, USA, 2007, pp. 1659–1664.
- [4] E. Seo, J. Jeong, S. Park, and J. Lee, "Energy efficient scheduling of real-time tasks on multicore processors," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 11, pp. 1540–1552, Nov. 2008.
- [5] A. Merkel and F. Bellosa, "Balancing power consumption in multiprocessor systems," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 4, pp. 403–414.
- [6] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical Data," Electrical Eng. Dept., Princeton Univ., Princeton, NJ, USA, Tech. Rep. EE-2003-09, Dec. 2003.
- [7] N. Kavvadias, P. Neofotistos, S. Nikolaidis, C. A. Kosmatopoulos, and T. Laopoulos, "Measurements analysis of the software-related power consumption in microprocessors," *IEEE Trans. Instrum. Meas.*, vol. 53, no. 4, pp. 1106–1112, Aug. 2004.
- [8] C.-H. Lien, Y.-W. Bai, and M.-B. Lin, "Estimation by software for the power consumption of streaming-media servers," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 5, pp. 1859–1870, Oct. 2007.
- [9] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.
- [10] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," *SIGARCH Comput. Archit. News*, vol. 38, no. 3, pp. 338–347, 2010.
- [11] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2001, pp. 1098–1106.
- [12] W. Dargie and A. Schill, "Analysis of the power and hardware resource consumption of servers under different load balancing policies," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, Washington, DC, USA, 2012, pp. 772–778. [Online]. Available: <http://dx.doi.org/10.1109/CLOUD.2012.30>
- [13] C. Mobius, W. Dargie, and A. Schill, "Power consumption estimation models for processors, virtual machines, and servers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 99, no. PrePrints, p. 1, 2013, DOI: 10.1109/TPDS.2013.183
- [14] S. Rivoire, "Models and Metrics for Energy-Efficient Computer Systems," Phd Dissertation, Stanford Univ., Stanford, CA, USA, 2008.
- [15] F. Bellosa, "The benefits of event: Driven energy accounting in power-sensitive systems" in *Proc. 9th Workshop ACM SIGOPS Eur. Workshop: Beyond the PC: New Challenges Oper. Syst.*, New York, NY, USA, 2000, pp. 37–42.
- [16] W. Bircher, "Complete system power estimation using processor performance events," *IEEE Trans. Comput.*, vol. 61, no. 4, pp. 563–577, Apr. 2012.
- [17] R. Bertran, M. G. Tallada, X. Martorell, N. Navarro, and E. Ayguade, "A systematic methodology to generate decomposable and responsive power models for CMPs," *IEEE Trans. Comput.*, vol. 62, no. 7, pp. 1289–1302, Jul. 2013.
- [18] G. Chen, W. He, J. Liu, S. Nath, and L. Rigas, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in *Proc. 5th USENIX Symp. Netw. Syst. Design Implementation*, San Francisco, CA, USA, pp. 337–350.
- [19] K. Singh, M. Bhadauria, and S. a. McKee, "Real time power estimation and thread scheduling via performance counters," *ACM SIGARCH Comput. Arch. News*, vol. 37, no. 2, pp. 46–55, Jul. 2009.
- [20] A. Lewis, S. Ghosh, and N.-F. Tzeng, "Run-time energy consumption estimation based on workload in server systems," in *Proc. Conf. Power Aware Comput. Syst.*, 2008, p. 4.
- [21] G. Dhiman, K. Mihic, and T. Rosing, "A system for online power prediction in virtualized environments using Gaussian mixture models," in *Proc. 47th Design Autom. Conf.*, New York, NY, USA, 2010, pp. 807–812.
- [22] R. Gray, "Gauss mixture vector quantization," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2001, vol. 3, pp. 1769–1772.
- [23] X. Fan, W.-d. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 13–23, Jun. 2007.
- [24] M. Pedram and I. Hwang, "Power and performance modeling in a virtualized server system," in *Proc. 39th Int. Conf. Parallel Process. Workshops*, 2010, pp. 520–526.
- [25] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*. 4th ed. New York, NY, USA: McGraw Hill, 2002.
- [26] R. Bertran, M. Gonzalez, X. Martorell, N. Navarro, and E. Ayguade, "Decomposable and responsive power models for multicore processors using performance counters," in *Proc. 24th ACM Int. Conf. Supercomput.*, New York, NY, USA, pp. 147–158.
- [27] N. Levinson, "The Wiener RMS (root mean square) error criterion in filter design and prediction," *J. Math. Phys. Mass. Inst. Tech.*, vol. 25, pp. 261–278, 1947.
- [28] D. Allen, "Mean square error of prediction as a criterion for selecting variables," *Technometrics*, vol. 13, no. 3, pp. 469–475, 1971.
- [29] T. Heath, B. Diniz, E. V. Carrera, W. M. Jr., and R. Bianchini, "Energy conservation in heterogeneous server clusters," in *Proc. 10th ACM SIGPLAN Symp. Principles Practice Parallel Programm.*, New York, New York, USA, pp. 186–195.
- [30] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments," in *Proc. 2nd Workshop Modeling, Benchmarking, Simul.*, Boston, MA, USA, 2006, pp. 70–77.
- [31] J. C. Mccullough, Y. Agarwal, J. Chandrashekar, S. Kuppaswamy, A. C. Snoeren, R. K. Gupta, and I. Labs, "Evaluating the effectiveness of model-based power characterization," in *Proc. USENIX Conf. USENIX Annu. Tech. Conf.*, Berkeley, CA, USA, 2011, p. 12.
- [32] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *Proc. IEEE/ACM Int. Symp. Microarchit.*, Dec. 2003, pp. 93–104, doi: 10.1109/MICRO.2003.1253186



**Walteneus Dargie (SM'12)** received the BSc degree from the Nazareth Technical College, Ethiopia, and the MSc degree from the Technical University of Kaiserslautern, Germany, in 1997 and 2002, respectively, both in electrical engineering. He received the PhD degree in computer engineering from the Technische Universität Dresden in 2006. He is an associate professor at the Technische Universität Dresden. He is a member of the editorial board of the *Journal of Network and Computer Applications*. His research interest is related to wireless sensor networks, stochastic processes, and energy-efficient computing. He is a senior member of the IEEE.