Run-time Energy Consumption Estimation Based On Workload in Server Systems

Adam Lewis[*], Soumik Ghosh[+], and N.-F. Tzeng[*]

[*]Center for Advanced Computer Studies, University of Louisiana, Lafayette, LA 70504, USA, {awlewis,tzeng}@cacs.louisiana.edu
[+]Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005, USA, soumik.ghosh@rice.edu

Abstract—**This paper proposes a system-wide energy consumption model for servers using hardware performance counters and experimental measurements. We develop a real-time energy prediction model that relates server energy consumption to its overall thermal envelope. While previous studies have attempted system-wide modeling of server power consumption through subsystem models, our approach is different in that it creates a model relating system energy input to subsystem energy consumption based on a small set of tightly correlated parameters. We develop a linear regression model that relates processor power, bus activity, and system ambient temperatures into real-time predictions of the power consumption of long jobs (and as result controlling their thermal impact). Using the HyperTransport bus model as a case study and through electrical measurements on example server subsystems, we develop a statistical model for estimating run-time power consumption. Our model is accurate within an error of four percent(4%) as verified using a set of common processor benchmarks.**

Index Terms—**Performance of systems, Measurement techniques, Modeling techniques, Large and Medium Computers.**

I. Introduction

THe upwardly spiraling operating costs of the infrastructure for enterprise-scale computing demand efficient power management in server environments. It is difficult in practice to achieve efficient power management as data centers usually over-provision their power capacity to address worst case scenarios. This results in either waste of considerable power budget or severe under-utilization of capacity. Thus, it is critical to quantitatively understand the relationship between power consumption and thermal load at the system level so as to optimize the use of deployed power capacity in the data center.

Power management techniques developed for mobile and desktop computers have been applied with some success to managing the power consumption of microprocessors used in server hardware. The current generation of Intel and AMD processors use different techniques for processor-level power management including (1) per core clock gating,(2) power-gating functional blocks (processors turn off certain blocks that are not in use) (3) multiple clock domains, (4) multiple voltage domains for cores, caches, and memory, (5) dynamic voltage and frequency scaling per core and processor, and (6) hardware support for virtualization techniques. In general, these techniques take advantage of the fact that reducing switching activity within the processor reduces energy consumption and application performance can be adjusted to utilize idle time on the processor for energy savings [1].

This paper presents a statistical full-system model that provides run-time system-wide prediction of energy consumption on server blades. Our model considers a single server blade as a closed black-box system. The black-box system model lets us converge upon an upper bound of the thermal, energy, and power envelopes of the system. We develop our model by measuring the energy input into the system as a function of the work done by the system in executing its computational tasks and residual thermal energy given off by the system in doing that work. A hardware performance counter (PeC) based relationship between server blade power consumption and the consequent thermal envelope is necessary to dynamically control the thermal footprint of large workloads. It is important to note that we are trying to establish an energy relationship between the workload and the overall thermodynamics of the system.

We begin by measuring the total DC power input to the system at the output from power supply. We partition the energy delivered to the system as a sum of the energies consumed by the different sub-systems in the server blade. We measure the computational load in the system by observation of the bus transactions that occur in the system as reported through the PeCs combined with a set of system metrics. These metrics are combined into a single model estimated through linear regression.

This work demonstrates that appropriate provision of additional PeCs beyond what are provided by a typical processor is required to obtain more accurate prediction of system-wide energy consumption. The model takes into account key thermal indicators and system parameters such as ambient temperatures, die temperatures, and hardware performance counters as metrics for system energy consumption estimation within a given power and thermal envelope.

We perform a case study of electrical measurements of a server architecture based upon the HyperTransport [2] bus model to develop our model to estimate run-time power consumption. Scheduler-based mechanisms are being developed to take advantage of this estimation model when dispatching jobs to confine server power consumption within a given power budget and thermal envelope while minimizing impact upon server performance.

II. Prior Work

Power models have been used to predict invocation of power management mechanisms in running server systems. These models can be classified into two broad categories: simulation-based models and detailed analytical power models. Although simulations can provide detailed analysis and breakdown of energy consumption, they are usually statically done off-line, are slow, do not scale well, and do not apply favourably to realistic applications and large data sets. In general, neither of these approaches have taken thermal effects of power dissipation into account. Management of system-critical thermal issues due to excessive power consumption, is further complicated by the existence of multiple cores per processor. However these simplistic simulation-based models do not fit well in scenarios where dynamic power and thermal optimization for application performance is required [3].

Analytical models use detailed knowledge of the underlying hardware to directly measure energy consumption at the hardware level. Measurement-based models attempt to collate power measurements to the micro-architectural units on the processor via sampling of with hardware and software performance metrics. Two distinct classes of metrics have been used in these models: processor performance counters and operating system performance metrics. Processor performance counters are hardware registers that can be configured to count various micro-architectural events such as branch mispredictions and cache misses. This type of model does not take account the energy consumption of devices other than the processor and rely upon detailed knowledge of the micro-architecture of the processor. Attempts have been made to reconcile these approaches by attempting to map programs phases to events [4]. The most common technique used to associate PeCs and/or operating systems metrics to energy consumption use linear regression models to the collected metrics to the energy consumed during the execution of a program [1][3][5][6].

In general, the number of recordable events exceed the number of available registers. As a result, models that use these counters time-multiplex different sets of events on the available registers. While this allows for more events to be monitored, it results in increased overhead and lower accuracy due to sampling issues [3][7]. It is critical that a power model be based upon the smallest possible set of metrics (either PeC or operating system metric) required to accurately model the system behavior in order to avoid the need for time-multiplexing. High level black-box models sacrifice some accuracy by avoiding extensive detailed knowledge of the underlying hardware. At the processor level, Contreras *et.al.* [1], and Bellosa [8] created power models that linearly correlated power consumption with performance counters. Models have been built for the processor, storage devices, single systems, and groups of systems in data centers. These models have the advantage of being simple, fast and low-overhead but do not model the full-system power consumption.

In server environments, it has been shown that full-system models using operating system CPU utilization can be highly accurate [9]. Others have used similar approaches [10] to develop linear models for energy-aware server consolidation in clusters. Full-system models such as MANTIS [3][7] relate usage information to the power of the entire system rather than an individual component. Each of these cases requires one or more calbibration phases that evaluates the contribution of each system component to overall power consumption. The accuracy and portability of full system power models is considered in Rivoire *et.al.* [11]. This analysis indicated that to ensure reasonable accuracy across machines and workload required a model based on a combination of both PeCs and operating system metrics, which directly accounted for all components of a system's dynamic power, and provided some insight into memory and disk power consumption.

Extensive study of the power profiles of the Intel Pentium architecture has occurred at the workstation [4] [5] [12] and server [6] [13] [14]. However, very little consideration has been given to the power profiles of servers constructed using the NUMA-based architecture used on processors such as the AMD64 family of processors [15].

III. System Model Design

We start by considering the type of power supplied into the system. Most server blades operate on an AC input. The DC output from the power supply in our experimental platform is delivered in the domains of +/-12V, +/-5V, and +/-3.3V [16]. In the case of the Sun server used in this study, two 12 Vdc lines supply power to the processor's hard drive(s) and cooling fans in the system. The 5 Vdc and 3.3 Vdc lines are dedicated to supplying power to the support chips and peripherals on the board. Most switched mode power supplies for servers have a power conversion efficiency from AC to DC of 72 - 80 % , depending on the load of the system. Typically for a server that is idling, i.e. is running the operating system and no other jobs, the power consumption is close to 40 - 42% of the rated power of the system (in our case 450W). The conversion efficiency increases to about 80% when heavily loaded , and the SMPS regulates the power supply to work at 75% conversion efficiency at loads over 50% of the rating.

Hence we can see that at the very outset that we lose 20% of the power supplied into the system to conversion losses even for the best conversion factors. Studies have shown [17] that most DC systems perform better in terms of power efficiency than AC systems. A typical AC-based server system has a power supply efficiency of 73% as compared to a 92% for a DC system. Also, the overall system efficiency for a AC system is 61% as compared to 85% for a DC system.

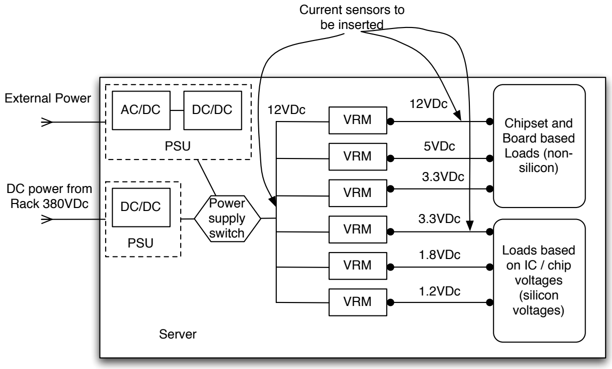
Current sensors to be inserted
External Power
AC/DC
DC/DC
PSU
DC power from Rack 380VDc
DC/DC
PSU
Power supply switch
12VDc
VRM
VRM
VRM
VRM
VRM
VRM
12VDc
5VDc
3.3VDc
3.3VDc
1.8VDc
1.2VDc
Chipset and Board based Loads (non-silicon)
Loads based on IC / chip voltages (silicon voltages)
Server

Fig. 1. Power distribution model for the server blade

Therefore, a rack level DC power distribution system easily translates into large power savings at the server level. We only consider a part of this power conversion unit as part of our model as it is not software tunable in its present state at the level considered within this work. However, current sensors, for example using MAXIM's 4473 [18], at the outputs from the power supply as performance counters would immensely aid in dynamically tracking DC power draw into the system which varies according to the system load. Our aim through this model is to monitor this input power to the system and control the power and consequently thermal envelope of the system based on the processing load. A proposed system diagram for a combined AC and DC power-supply based system with performance counter measurable current sensors, along with power distribution, is shown in Fig. 1.

In order to develop an energy consumption model based on computational load of the system, we begin by measuring the total DC power input to the system, at the output of the SMPS. As mentioned earlier, the DC power is delivered in domains of +/- 12 V, +/-5V, and +/- 3.3V. Most SMPS will limit the total power delivered through the 5V and 3.3V lines to about 20% of the rated power supply (P_R). Now assuming each of the voltage lines $v_k(t)$ draws current $i_k(t)$, then each line draws an instantaneous power $p_k(t) = v_k(t) \cdot i_k(t)$. If a voltage domain has M DC lines as output, then total power delivered for that voltage domain is :

$$p_{v1}(t) = \sum_{k=0}^{M} v_k(t) \cdot i_k(t)$$

If the board has N voltage domains, then the total DC power delivered into the system is :

$$p_{dc}(t) = \sum_{j=0}^{N} p_{vj}(t) = \sum_{j=0}^{N} \sum_{k=0}^{Mj} v_k(t) \cdot i_k(t)$$

So total energy delivered to the system between times t_2 to t_1 is :

$$\begin{aligned} E_{dc} &= \int_{t_1}^{t_2} p_{dc}(t)dt \\ &= \int_{t_1}^{t_2} \sum_{j=0}^{N} p_{vj}(t)dt \\ &= \int_{t_1}^{t_2} \sum_{j=0}^{N} \sum_{k=0}^{Mj} v_k(t) \cdot i_k(t)dt \end{aligned}$$

For the 3.3V and 5V lines thus, the following constraint holds :

$$\begin{aligned} E_{dclv} &= \int_{t_1}^{t_2} p_{dclv}(t)dt \\ &= \int_{t_1}^{t_2} (\sum_{k=0}^{M1} v_k(t) \cdot i_k(t) + \sum_{k=0}^{M2} v_k(t) \cdot i_k(t))dt \\ &\leq 0.2P_R \end{aligned}$$

where $M1$ and $M2$ are the total 3.3V and 5V lines respectively. Thus in our 450W rated system the power delivered by the 3.3V and 5V lines is capped at 90W.

This energy delivered to the system $E_{dc} = E_{system}$ can now be expressed as a sum of energies consumed by the different sub-systems in the server blade. Broadly we define five sources of energy consumption within a system:

- E_{proc}: Energy consumed in the processor due to all computations
- E_{memory}: Energy consumed in the DRAM chips and
- E_{em}: Energy consumed by all electrical and electromechanical components in the server blade. This includes fans, and other components on the server which consume AC power.
- E_{board}: Energy consumed by perphierals that support the operation of the board. These include all devices in the multiple voltage domains across the board, in cluding chipset chips, voltage regulation, bus control chips, connectors, interface devices etc.
- E_{hdd}: Energy consumed by the hard disk drive during the server's operation.

We explore each of these terms in turn by following an energy conservation model in the system. In order to a get a true measure of the computational load on the system, our method looks to snoop on completed bus transactions per unit time in the system and measure the relative change in energy consumption (as indicated by change in temperature) as computation tasks are completed. Use of this performance counter metric as compared to other metrics fits well with the architecture of microprocessors used in NUMA-based processors in multi-core environments.

Consider the Intel Pentium and AMD Operton processor architectures connected in a dual core configuration shown in Fig. 2 and 3. The Pentium architecture (and its successors) is based upon the idea of a Front-Side-Bus (FSB) which connects individual cores to the Northbridge chip. This chip provides the interface between the cores and memory. A coherent bus protocol is used to ensure consistency in memory access between the cores. For this architecture, the FSB becomes a performance bottleneck as processor and cores have to moderate themselves to the slower speed of the bus.

Contrast this with the NUMA-based architecture used by the AMD Opteron (Fig. 3). In this case, the Northbridge functionality is combined onto the same processor die as each core and each core is responsible for local access to the memory connected to that Northbridge. Cores on a single die are connected via a crossbar to the Hypertransport bus between processors. Again, a coherent bus protocol is used to ensure memory consistency between cores and processors. In addition, the master processor in the system is connected via a second Hypertransport bus to the Southbridge device that manages connections to the outside world.

We see that the work done by any of these processors, which is at the heart of energy consumption in a server system, can be quantified in terms of bus transactions in and out of these processors. The traffic on the external buses give us a measure of

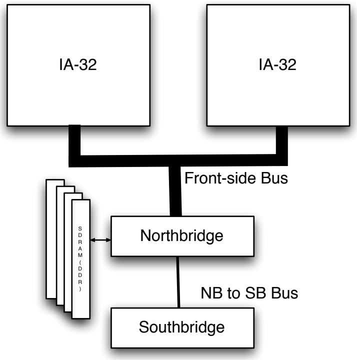

Fig. 2. Intel Core Server Architecture

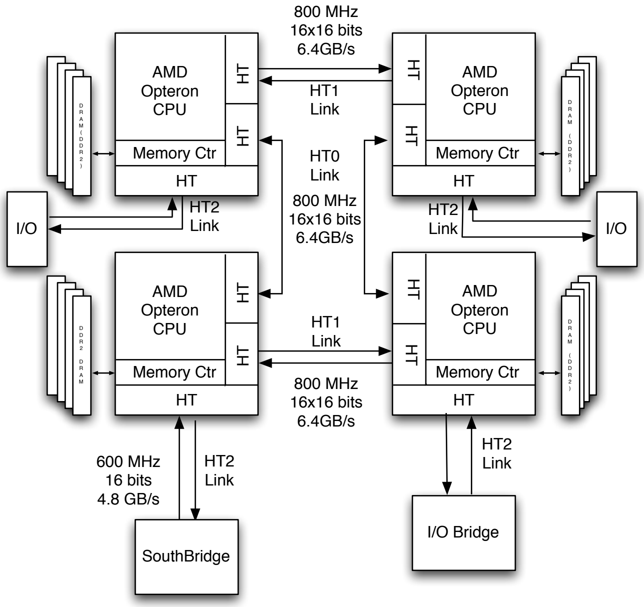

Fig. 3. AMD Opteron Server Architecture

how much data is being processed by the processor and what would be an upper limit of the work done by a single processor. In our approach we concentrate on developing the energy consumption model on a Hypertransport (HT) based system with two AMD dual-core Opterons in a Sun X2200 server [19].

A. Processor energy consumption

Our processor model aims to treat the processor as a black box, whose energy consumption is a function of its work load, and the work done manifests as the core die-temperature and system ambient temperature (measured at a system level by `ipmitool` through sensors in the path of the outgoing airflow from the processor). A practical issue with trying to estimate processor power using a large number of PeCs is that there are only a limited number of PeCs that tools like `cpustat` can track simultaneously. In order to track the energy-thermal load relationship for a job, we had to develop a model with the least number of PeCs that would accurately reflect the energy consumption-thermal load relationship.

Given the AMD Operton processor architecture connected in a dual core configuration shown in Fig. 4, we consider traffic on the HyperTransport buses as representative of the processor work load and reflecting the amount of data being processed by a processor or any of its cores. The HT2 bus is non-coherent and connects one of the two processors to the Southbridge (whereas the Northbridge is inside the Opteron processor). Thus, traffic on the HT2 bus reflects hard-disk and network traffic. The model therefore scales when considering the effect of network traffic and heavy disk I/O based jobs. HT1 is a coherent bus between the two SMP processors and PeCs on that bus to give an accurate reflection on the processing load of cores executing jobs. Per-core die temperature readings and, consequently, ambient temperature per processor are thus greatly affected by the number of transactions over the HT buses. We also include L2 cache misses as one of our variables (to be explained in Section III-B).

Thus the total processor power consumption to reflect the thermal change due to workload can be expressed as:

$$\mathbf{P_{proc}} = \mathbf{H} \cdot \mathbf{X} = [\beta_\mathbf{0} \cdots \beta_\mathbf{10}]^\mathbf{T} \cdot [\mathbf{Var_0} \cdots \mathbf{Var_{10}}]^\mathbf{T}$$

where $\mathbf{X}$ vector contains the following variables: ambient temperatures and die temperatures for processors 0 and 1, HT1 and HT2 transactions, and L2 cache misses per core. The popularity of HyperTransport in server and high performance computing platforms based on AMD, IBM, nVidia, Altera, and Cray processors makes the model applicable to a wide variety of platforms.

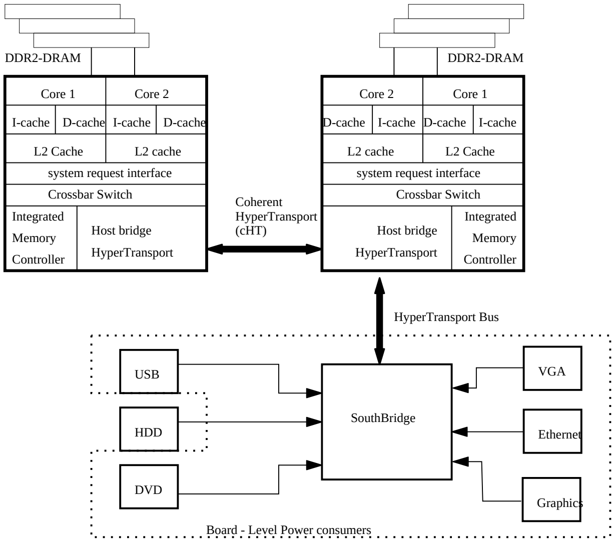

Fig. 4. Dual-core AMD Opteron based server architecture.

B. DRAM energy

Energy consumed by the DRAM banks can be computed by a combination of measuring the counts of the highest level cache miss in the processor combined with the DRAM Read/Write power along with the DRAM background power(activation power). As illustrated in [20], DRAM background power and activation power can be obtained from the DRAM datasheets. For a single DRAM in our case, a total of 493mW would be consumed. However, given the number of L2 cache misses per second when a job is running on a certain core (over 22M / sec at the peak of bzip2 SPEC2006 benchmark), a significant amount of heat is generated from the DRAM chips. The thermal airflow proximity of the DRAM banks to their respective processors makes it possible for us to combine the energy consumption and the consequent thermal output of the memory banks with the processor ambient temperature. This value is reported by IPMI and we combine it into our regression model.

C. Hard disk energy

The energy consumed by the hard disk while operating, can be approximated to give an upper bound on the energy consumption of the hard disk using a combination of performance counters and drive ratings. In our server, Hitcahi's 7200 RPM 250G, SATA hard disk is used. Table I lists the typical power consumption numbers for the hard disk used. Based on the physical, electrical and electromechanical parameters of the hard disk, very detailed power consumption models can be constructed. However we can achieve a cruder but simpler model based on the typical power consumption data of the hard disk and performance counters.

The utility `iostat` can be used to measure the number of read and writes per second to the disk as well as the kilobytes read from and written to the disk. Thus based on this performance counter, we can compute an approximate disk power consumption E_{hdd} value as :

$$\begin{aligned} E_{hdd} = & P_{spin-up} \times T_{su} \\ & + P_{read} \sum N_r \times T_r \\ & + P_{write} \sum N_w \times T_w \\ & + \sum P_{idle} \times T_{id} \end{aligned}$$

where $P_{spin-up}$ is the power required to spin-up the disk from 0 to full rotation. T_{su} is the time required to achieve spin up. T_{su} is typically about 10s. P_{read} is the power consumed per kilobyte of data read from the disk. N_r is the number of kilobytes of data read in time-slice T_r from the disk. The variables are analogous for the write energy consumption. P_{read} for our Hitachi disk can be computed as follows:

The read operation at 1.5 Gbits/s consumes 530 mA current at +5V. Hence every kilobyte read, consumes approximately $13.3\mu W$/Kbyte. Similarly, every write operation consumes $6.67\mu W$/Kbyte. The numbers N_r and N_w can be obtained using `iostat` and out choice of time-slice.

The idle state has two conditions, idle and unloaded idle, where in the latter case the heads are unloaded. The time to go from unloaded to idle is usually less than 1 second, which is less than the resolution of `iostat`. Thus, a history match count in the `iostat` statistics where the reads and writes have been zero, tells us the period in which the disk is idle, and the idle energy consumption can be computed accordingly. `iostat` reading based conditions for switching to different disk power states can be obtained with more in-depth analysis, but the net results falls into this equation's framework. That analysis is the topic of a future work.

The hard disk power can also be measured in real-time if current sensors are provided at the output of the DC voltage lines delivering power to the hard disk drives. The +5V lines will draw a maximum

TABLE I
HITACHI HDT725025VLA360 DISK POWER PARAMETERS

Parameter	Value
Interface	Serial ATA
Capacity	250 GB
Rotational speed	7200 rpm
Power (spin up)	5.25 W (max)
Power (Random read, write)	9.4 W (typical)
Power (Silent read, write)	7 W (typical)
Power (idle)	5 W (typical)
Power (low RPM idle)	2.3 W (typical for 4500 RPM)
Power (standby)	0.8 W (typical)
Power (sleep)	0.6 W (typical)

current of 730mA and the +12V lines will draw a maximum current of 630mA. Thus E_{hdd} can also be formulated as :

$$E_{hdd} = \int_{t1}^{t2} \{v_1(t) \times i_1(t) + v_2(t) \times i_2(t)\} \, dt$$

This approach can be applied in the presence of current sensors which in our experiments was measured with a current probe and logged through an oscilloscope.

D. Board Components

The quantity E_{board} captures the energy required by the support chipsets and usually fall in the 3.3V and 5V power domains. In our case we obtain this value using current probe based measurements. However, as in earlier cases, current sensors for the power lines going into the board can provide instantaneous energy draw from the power supply. The processor, disk, fan, and optical-drive power lines are excluded here. For our server, at most 28 additional current sensors might be required for the entire blade [16]. Thus:

$$E_{board} = \left(\sum V_{power-line} \times I_{power-line}\right) \times t_{time-slice}$$

E. Electromechanical Energy

There is a basic electrical cost related to running the computer. The quantity E_{elect} in our model takes these quantities into account. E_{elect} is calculated as summation of the DC and AC power consumption in the peripherals supporting the processor, particularly the electromechanical components. This is mainly the power cosumption in the power supply unit and the power consumption in the cooling fans. As discussed earlier, the AC to DC conversion process is a load based number and has a best case conversion efficiency of 80% and a nominal efficiency of 75% .

Power drawn by the fans for cooling can be given by the following equation:

$$P_{fan} = P_{base} \cdot \left(\frac{RPM_{fan}}{RPM_{base}}\right)^3$$

P_{base} defines the base power consumption of the unloaded system. In our case, that is the power consumption of the system when running only the base operating system and no other jobs. That value is obtained experimentally by measuring the current drawn on the +12V and +5V lines, using a current probe and an oscilloscope. There is a current surge at system startup, which is neglected and under nominal conditions, the +12V line draws approximately 2.2A, which powers both the blowers fans in the system. The two peripheral fans running at +5V draw around 2.1A of current. Thus the base power for the fans in known. IPMI sensors easily collect fan RPM data, and hence it is possible to quantify the electrical power consumption in the system. Thus the electrical power consumption can be quantified as:

$$P_{elect} = V(t) \cdot I(t) + \sum_{i=1}^{N} P_i$$

where the first term in the equation is the instantaneous DC power output from the power supply and is the DC power consumed by the system. N is the number of fans in the server and P_i is the

TABLE II
OVERALL REGRESSION MODEL.

Coeff.	Variable	Measurement
β_0		
β_1	T_{A_0}	Ambient Temp0
β_2	T_{A_1}	Ambient Temp1
β_3	T_{C_0}	CPU0 Die Temp
β_4	T_{C_1}	CPU1 Die Temp
β_5	HT_1	HT1 Bus X-Actions
β_6	HT_2	HT2 Bus X-Actions
β_7	CM_0	L1/L2 Cache Miss for Core0
β_8	CM_1	L1/L2 Cache Miss for Core1
β_9	CM_2	L1/L2 Cache Miss for Core2
β_{10}	CM_3	L1/L2 Cache Miss for Core3
β_{11}	D_r	Disk bytes read
β_{12}	D_w0	Disk bytes written
β_{13}	F_C	CPU Cooling Fan Speed
β_{14}	F_M	Memory Cooling Fan Speed

instantaneous power consumed by the $i - th$ fan according to equation III-E.

$$\eta = \frac{V(t) \cdot I(t)}{P_{in}}$$

η gives a measure of the energy conversion efficiency, into the system from the mains, and gives an idea of the energy budget available to the system.

Thus the total energy consumption during a given task period T_p due to electrical energy in the system can now be given by:

$$E_{elect} = \int_0^{T_p} [V(t) \cdot I(t) + \sum_{i=1}^{N} P_i] \, dt$$

F. Combined Model

The total energy consumed by the system for a given computational workload is modeled as a function of these metrics as:

$$E_{system} = \alpha_0(E_{proc}+E_{mem})+\alpha_1 E_{em}+\alpha_2 E_{board}+\alpha_3 E_{hdd}$$

where α_0,α_1,α_2, and α_3 are unknown constants that are determined through linear regression analysis and remain constant for any given server architecture.

IV. STATISTICAL SYSTEM MODEL

The empirical model is based on application data driven activity and data flow across the server system. The components described in the previous sections can only be formulated in a heuristic rather than analytical manner. This arises from the lack of measurement and monitoring capabilities and data flow state capture mechanisms within the system which aid in formulating an exact analytical model. Thus, we are required to derive a statistical fit model based upon experimental results as described in Section V. Our heuristical framework for system thermal sub-components leads us to the statistical parameters to be monitored and measured in deriving this model.

For example, the system board DC and AC power consumption cannot be easily split in terms of measurements or exact analytical models due to the presence of large numbers of voltage and current domains and components. However, the concept of AC and DC power consumption on the board is captured as voltage and current domain based power summation in our heuristical model. However, it is only through data-driven black box experiments that we can estimate the aggregate power consumption of each sub-component. It is important to note that these board level components only play a supportive role in energy consumption due to computation; however, they reflect the application's thermal load on the system.

A physical predictor was selected for each term in the analytical model (shown in Table II). Each predictor contributes to one of the terms in our combined model. For example, we predict that the energy consumed in the process of computation E_{proc} can be estimated by a linear combination of the CPU Die Temperatures and the amount of data transmitted across the HyperTransport bus connecting the physical cores:

$$E_{proc} \cong \beta_3 * T_{C_0} + \beta_4 * T_{C_1} + \beta_5 * HT_1$$

The energy consumed in the memory system E_{mem} is estimated by a combination of the amount of data transmitted across the Hypertransport bus between processor and the Southbridge and the number of last level cache misses for each physical core:

$$\begin{aligned} E_{mem} \cong \beta_6 * HT_2 + \beta_7 * CM_0 + \beta_8 * CM_1 \\ + \beta_9 * CM_2 * \beta_{10} * CM_3 \end{aligned}$$

We estimate the energy consumed by the electrical components in the server blade as a linear combination of the ambient temperature as measured in the air flow moving over each physical processor:

$$E_{em} \cong \beta_1 * T_{A_0} + \beta_2 * T_{A_1}$$

The energy consumed as result of the peripheral operation of the board is estimated as a linear combination of the operating speeds of the fans cooling the CPU and memory plus the linear estimation constant term:

$$E_{board} \cong \beta_0 + \beta_{13} * F_C + \beta_{14} * F_M$$

The energy consumed as result of disk activity is estimated by measuring the number of bytes read and written to the desk

$$E_{hdd} \cong \beta_{11} * D_r + \beta_{12} * D_w$$

Note the importance of the linear estimations coefficients in our statistical model. It is the units associated with each coefficient that is responsible for converting each quantity to map the units associated with each variable into energy units.

The predictors are combined together into a linear regression model:

$$\begin{aligned} E_{system} \cong \beta_0 + & \beta_1 * T_{A_0} + \beta_2 * T_{A_1} + \\ & \beta_3 * T_{C_0} + \beta_4 * T_{C_1} + \\ & \beta_5 * HT_1 + \beta_6 * HT_2 + \\ & \beta_7 * CM_0 + \beta_8 * CM_1 + \\ & \beta_9 * CM_2 + \beta_{10} * CM_3 \\ & \beta_{11} * D_r + \beta_{12} * D_w \\ & \beta_{13} * F_C + \beta_{14} * F_M \end{aligned}$$

This model is fit to a collection of representative benchmarks from the SPEC CPU2006 benchmark suite [21] as listed in Table III. The benchmarks were selected using two criteria: sufficient coverage of the functional units in the processor and reasonable applicability to the problem space. Components of the processor affect the thermal envelope in different ways [22]. This issue is addressed by balancing the benchmark selection between integer and floating point benchmarks in the SPEC CPU2006 benchmark suite. Second, the benchmarks were selected from the suite based upon fit into the problem space. Each benchmark represents an application typical of the problems solved on high-performance application servers.

Five classes of metrics are sampled at 5 second intervals during the experiment: (1) CPU temperature for all processors in the system, (2) Ambient temperature in the computer case measured in one more locations using the sensors provided by server manufacturer, (3) the number of completed transactions processed through the system bus, and (4) the number of misses that occur in the L2 cache associated with each CPU core in the system.

TABLE III
SPEC CPU2006 BENCHMARKS USED FOR MODEL CALIBRATION.

Benchmark		Type	Use
perlbench	C	Int	PERL Programming Language
bzip2	C	Int	Compression
mcf	C	Int	Combinatorial Optimization
omnetpp	C++	Int	Discrete Event Simulation
gromacs	C/F90	FP	Biochemistry/Molecular Dynamics
cacstusADM	C/F90	FP	Physics/General Relativity
leslie3d	F90	FP	Fluid Dynamics
lbm	C	FP	Fluid Dynamics

Two methods were considered for consolidation: arithmetic mean (average) and geometric mean. Trial models were constructed using each method and a statistical analysis of variance was performed to determine which model generated the best fit to the collected data. Fig. 5 and 6 provides a visual comparison of the system power as measured on the SUT device versus the predicted power consumption from each device.

V. Case Study

A case study of the use of our power model to evaluate energy consumption on a test system was created for the hardware described in Table V. The power consumed is measured with a WattsUP [23] power meter connected between the AC Main and the system under test (SUT). The power meter measures the total and average wattage, voltage, and amperage over the run of a workload. The internal memory of the power meter is cleared at the start of the run and the measures collected during the run are downloaded after the run completes from the meter's internal memory into a spreadsheet [24]. Current flow on the different voltage domains in

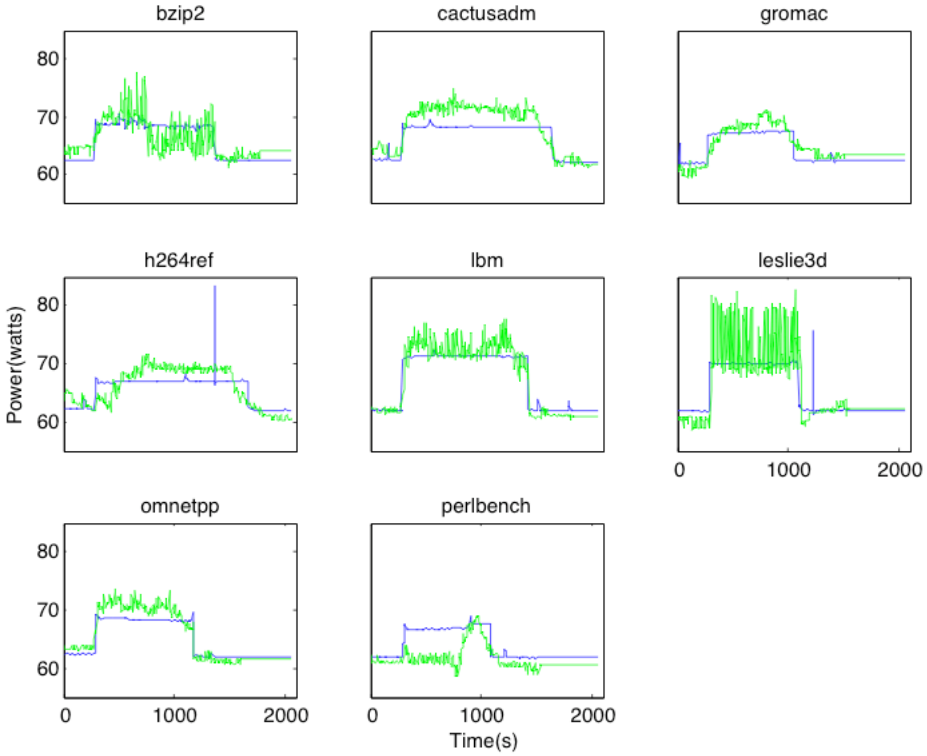

Fig. 5. Actual energy vs. predicted (geometric mean)

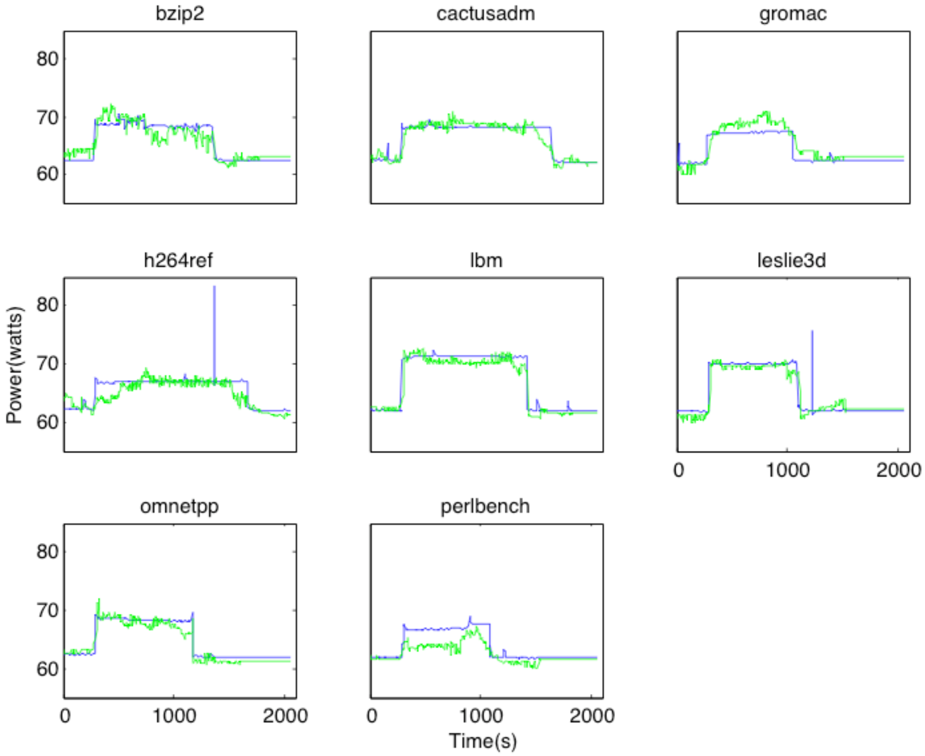

Fig. 6. Actual energy vs. predicted (arithmetic mean)

the server is measured using an Agilent MSO6014A oscilloscope with one Agilent 1146A current probe per system power domain (12v, 5v, and 3.3v). This data is collected from the oscilloscope at the end of the execution of a benchmark and stored in a spreadsheet on the test host.

TABLE IV
TEST HARDWARE CONFIGURATION.

	Sun Fire 2200
CPU	2 AMD Opteron
CPU L2 cache	2x2MB
Memory	8GB
Internal disk	2060GB
Network Interface Card	2x1000Mbps
Video	On-board
Height	1 rack unit

The operating system used in our setup is OpenSolaris (Solaris 11). System data is collected from the system baseboard controller using the IPMI interface using the OpenSolaris `ipmitool` utility. Processor performance counters are collected on a system-wide basis using the OpenSolaris `cpustat` utility.

A linear regression model was created from the consolidated data set to generate the parameters described in our theoretical model:

$$\begin{aligned} E_{system} \cong 32.71 + {} & 1.31 * T_{A_0} + 0.54 * T_{A_1} + \\ & 0.54 * T_{C_0} + 0.61 * T_{C_1} + \\ & 0.01 * HT_1 + 0.01 HT_2 + \\ & 0.01 * CM_0 + 0.01 * CM_1 + \\ & 0.01 * CM_2 + 0.01 * CM_3 \\ & 0.50 * D_r + 0.50 * D_w \\ & 0.01 * F_C + 0.01 F_M \end{aligned}$$

The fit of this model to the consolidated data was done through an Analysis of Variance (ANOVA) and related statistical tests of the parameters and overall model fit (shown in Table V). An adjusted R-Square value of 0.965 gives an projected error for the model of 3.5 percent with a 95% confidence level.

TABLE V
ANOVA FOR CONSOLIDATED MODEL.

Source	df	SS	MS	F	P
Regr	10.00	2261.17	226.12	1150.23	0.00
Resid	399.00	78.44	0.1966		
Total	409.00	2339.60			
R-sq	0.97	Adj. R-sq	0.97		

TABLE VI
SPEC CPU2006 BENCHMARKS USED FOR EVALUATION.

Benchmark		Type	Use
astar	C++	Int	Path Finding
gobmk	C	Int	Artificial Intelligence: Go
calculix	C++ & F90	FP	Structural Mechanics
zeusmp	F90	FP	Computational Fluid Dynamics

A set of four additional benchmarks from the SPEC CPU2006 (shown in Table VI) was executed to evaluate the predictive performance of the model. We compare the predicted power consumption from the model versus the actual power consumption for each benchmark. Table VII shows the descriptive statistics and percentage error for the predictions for benchmark.

VI. EVALUATION

The consolidated model is attempting to predict for all benchmarks. Given the large volume of data generated thorough the different logging mechanisms, it is nearly impossible to discard bad data. Using the geometric mean as discussed in the previous section helps to smooth out some of the errors introduced in the cases. However, the diversity of the benchmarks used means that some discrepancies arise within variables where we expect to see tight correlations. Thus, the model predicts well in some cases and not in others. The worst error is no more than the four present reported above.

Also, the asymmetry of the β-coefficients for tightly correlated variables (the variables for the HyperTransport bus HT_1 and HT_2, for example) leads us to believe non-linear relationships may exist among these variables. Therefore, future work needs to consider the impact of use of non-linear models regression models together with hardware performance counters.

Another observation from the model pertains to the placement of the temperature sensors in the server. The ambient temperature regression variable T_{A_0} reflects more of the hot air flow due to the server design. This illustrates that for different server designs the factors controlling the thermal envelope

TABLE VII
MODEL ERROR FOR EACH BENCHMARK.

Benchmark	Mean	Median	Std	Var	Percent Error
astar	1.1172	0.9781	0.9533	0.9087	1.9%
calculix	1.3833	1.1895	1.2372	1.5306	2.3%
gobmk	2.3039	2.2006	0.8175	0.6684	3.9%
zeusmp	1.8336	1.7587	1.1333	1.284423	3.1%

TABLE VIII
MODEL ERROR FOR A MODEL BUILT FROM ONLY INTEGER BENCHMARKS.

Benchmark	Mean	Median	Std	Var	Percent Error
astar	18.5668	17.9879	2.1955	4.8201	3.2%
calculix	17.7971	16.3383	6.8716	47.2186	3.0%
gobmk	15.0118	13.8667	2.4372	5.9398	2.5%
zeusmp	14.1532	13.6791	2.9838	8.9028	2.4%

will be accurately reflected in the model. Thus, we would expect to see a more symmetric set of coefficients for ambient temperature variables T_{A_0} and T_{A_1} had the placement of the sensors been more balanced in the server.

A. Choice of Benchmarks

The benchmarks used to calibrate and evaluate the model were chosen based on two factors: instruction mix and workload type. Benchmarks were selected in an attempt to balance between integer and floating-point instructions. Follow-up tests were performed to evaluate the impact of instruction mix on the predictive ability of the model. Predictive models were created using only the integer benchmarks and only the floating-point benchmarks listed in Table III. In Table VIII and Table IX, we use each of these models to predict the energy consumption for each of benchmarks in our study. We see that the overall model is a better predictor for the general case as opposed to using the integer-only or floating point-only models. In future work, we will consider benchmarks focused on disk-use or memory utilization.

B. Measurement Tools

In terms of measuring performance counters, we have used the OpenSolaris `cpustat`, `iostat`, and `ipmitool` utilities. Of these, `iostat` and `ipmitool` are available across all UNIX-based operating systems commonly used in data centers. `cpustat` is an OpenSolaris specific utility but is already being ported to Linux. In future work, it is planned to use tools like `dtrace` and `oprofile` for more controllable and tunable performance parameters which have major impacts on system-wide and processor wide power consumption.

C. Processor Platform

Though the computations have not yet been done on Intel's Xeon platforms, the computation methodology holds for those platforms. We would measure Xeon performance counters like `BUS_TRANS_ANY`, `BUS_TRANS_MEM`, and `BUS_TRANS_BURST`. Similar model development and coefficient extraction arguments would hold for dual and quad core Xeons in different processor configurations. Currently without data from the Intel processors it is hard to say whether the model is more accurate on a certain platform as compared to the other.

For a practical usage scenario the statistical coefficients need to be computed only once using the SPEC benchmarks for a given server architecture. They can be used as embedded constants available either through the system firmware or the operating system kernel.

The model developed in this paper is valid for any AMD Opteron dual-core/dual-processor system

TABLE IX
MODEL ERROR FOR A MODEL BUILT FROM ONLY FLOATING POINT BENCHMARKS.

Benchmark	Mean	Median	Std	Var	Percent Error
astar	5.2434	5.1023	1.7213	2.9627	8.9%
calculix	3.9960	4.2091	2.1481	4.6146	6.7%
gobmk	4.8516	3.7065	2.4371	5.9399	8.2%
zeusmp	3.5565	3.2692	1.7648	3.1147	6.0%

using the HyperTransport system bus. However, it is scalable to any quad-core dual processors Opteron system using HyperTransport. One would expect to see a slight difference or variation in the predicted power due to the greater or diminished affect of the die temperatures on the other parameters and the model would have to be adjusted accordingly. For a dual-core quad-processor system, the additional term HyperTransport bus regresion variable HT_0 would be introduced into the CPU power consumption term and the β coefficients would have to be recalculated and the CPU power equation will have more terms. For a quad-core dual-processor system, similar recalculations would be required.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a comprehensive model which uses statistical methods to predict system-wide energy consumption on server blades. The model measures energy input to the system as a function of the work done for completing tasks being gauged and the residual thermal energy given off by the system as a result. Traffic on the system bus, misses in the L2 cache, CPU temperatures, and ambient temperatures are combined together using linear regression techniques to create a predictive model which can be employed to manage the processor thermal envelope.

The experimental validation of our model reveals opportunities for further investigation. The model has been validated for NUMA-based systems such as the AMD Operton processor; it requires validation on other architectures such as the Intel Xeon and IBM Cell BE processors. Further study of the power and thermal envelope of non-CPU components (memory and disk) to better understand their contributions to the system thermal envelope.

A fast, accurate, and robust model for the power and thermal envelope for a single server blade is critical to understanding and solving the power management challenges unique in dense servers. The model presented in this work is the first step towards building solutions that bridge multi-core, multiple blade, and full data center power management.

REFERENCES

[1] G. Contreras and M. Martonosi, "Power Prediction for Intel XScale®Processors Using Performance Monitoring Unit Events," in *ISLPED '05: Proc. of the 2005 Intl. Symp. on Low Power Electronics and Design*. New York, NY, USA: ACM, 2005, pp. 221–226.

[2] H. T. Consortium, "HyperTransport I/O Link Specification," HyperTransport Technology Consortium, Specification 3.00c, September 2007.

[3] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-System Power Analysis and Modeling for Server Environments," in *Workshop on Modeling Benchmarking and Simulation (MOBS) at ISCA*, 2006.

[4] C. Isci and M. Martonosi, "Phase Characterization for Power: Evaluating Control-flow-based and Event-counter-based Techniques," *The 12th Intl. Symp. on High-Performance Computer Architecture*, pp. 121–132, 11-15 Feb. 2006.

[5] ——, "Runtime Power Monitoring in High-end Processors: Methodology and Empirical Data," *MICRO-36: Proc. 36th IEEE/ACM Intl. Symp. on Microarchitecture*, pp. 93–104, 3-5 Dec. 2003.

[6] W. Bircher and L. John, "Complete System Power Estimation: A Trickle-Down Approach Based on Performance Events," *Ispass*, vol. 0, pp. 158–168, 2007.

[7] S. Rivoire, "Models and Metrics for Energy-efficient Computer Systems," Ph.D. dissertation, Stanford University, 2008.

[8] F. Bellosa, A. Weissel, M. Waitz, and S. Kellner, "Event-driven energy accounting for dynamic thermal management," *Proceedings of the Workshop on Compilers and Operating Systems for Low Power (COLP'03)*, 2003.

[9] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ISCA '07: Proc. of the 34th Intl. Symp. on Computer architecture*. New York, NY, USA: ACM, 2007, pp. 13–23.

[10] T. Heath, B. Diniz, E. V. Carrera, W. M. Jr., and R. Bianchini, "Energy Conservation in Heterogeneous Server Clusters," in *PPoPP '05: Proc. of the 10th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming*. New York, NY, USA: ACM, 2005, pp. 186–195.

[11] S. Rivoire, P. Ranganathan, and C. Kozyrakis, "A Comparison of High Level Full-System Power Models," in *Proc. of USENIX Workshop on Power Aware Computing and Systems (HotPower'08)*, 2008.

[12] C. Isci and M. Martonosi, “Identifying Program Power Phase Behavior Using Power Vectors,” *WWC-6:IEEE 2003 Intl. Workshop on Workload Characterization*, pp. 108–118, 27 Oct. 2003.

[13] W. Bircher, J. Law, W. Valluri, and L. John, “Effective Use of Performance Monitoring Counters for Run-Time Prediction of Power,” The University of Texas at Austin, Tech. Rep. TR-041104-01, November 2004.

[14] K.-J. Lee and K. Skadron, “Using performance counters for runtime temperature sensing in high-performance processors,” *Proc. 19th IEEE Intl. Symp. Parallel and Distributed Processing*, pp. 8 pp.–, 4-8 April 2005.

[15] *AMD Opteron Processor Data Sheet*, 3rd ed., AMD, March 2007.

[16] “EPS12v Power Supply Design Guide, V2.92,” Server System Infrastructure Consortium, Spec. 2.92, 2004.

[17] M. Ton, B. Fortenbery, and W. Tschudi. DC Power for Improved Data Center Efficiency.

[18] Maxim. Practical Considerations for Advanced Current Sensing in High-Reliability Systems.

[19] Sun Microsystems, Inc, *Sun Fire X2100 M2 and X2200 M2 Server Architecture*, white paper, Sun Microsystems, Inc, April 2008.

[20] “Calculating Memory System Power for DDR3,” Micron, Inc, Tech. Note TN41_01DDR3 Rev.B, August 2007.

[21] J. L. Henning, “Spec cpu2006 benchmark descriptions,” *Computer Architecture News*, vol. 34, no. 4, Sept 2006.

[22] A. Kumar, L. Shang, L.-S. Peh, and N. Jha, “System-Level Dynamic Thermal Management for High-Performance Microprocessors,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 96–108, Jan. 2008.

[23] Electronic Educational Devices, Inc., “WattsUp Power Meter,” December 2006.

[24] I. Electronic Educational Devices, “WattsUp Communication Protocol.”