Modeling the Power Consumption of Computer Systems with Graphics Processing Units (GPUs)

Stephanie Schmidt
Department of Computer Science
Sonoma State University
stphschmidt@gmail.com

ABSTRACT

To help optimize computer systems' energy efficiency, researchers have developed models relating these systems' instantaneous power consumption to resource utilization metrics available in software. Existing power models assume the major consumers of dynamic power are the central processing unit (CPU), memory and disk.

The increased use of graphics processing units (GPUs) for general-purpose computing compromises this assumption. Therefore, models that exclude the GPU may significantly mispredict power consumption. I built full-system power models for two GPU-equipped machines in order to test this hypothesis.

This research is the first to quantify: the inaccuracy of traditional high-level system power models for GPU workloads; the benefits of including GPU metrics in a full-system power model when applied to both GPU and non-GPU workloads; and the sensitivity of the model to the specific choice of GPU metrics.

Categories and Subject Descriptors

C.4.d [**Computer Systems Organization**] Performance of Systems – *modeling techniques*

General Terms

Measurement, Performance

1. PROBLEM AND MOTIVATION

Energy consumption is a major concern in computer systems [1, 8]. In personal computing, energy efficiency is important because no one enjoys having their mobile device die from one too many games of Angry Birds. Growing in scale, data centers and supercomputing centers have a laundry list of energy-related concerns: power and cooling costs, power density and heat, pollution and the load on the facility's utility provider. When data centers that house thousands of computers (and can be the size of multiple football fields) go over the power allotment for the building, the companies that operate them are left with two options: become energy efficient or build a whole new data center. The latter option is undesirable for the obvious reason: how long would this new costly facility last if energy demands keep increasing?

Because power and energy have become so important, component manufacturers have responded by introducing different operating modes that trade off power with performance. For example, most modern CPUs are able to operate at multiple frequencies, depending on whether the user prioritizes speed or power savings.

There is a large body of research investigating how to use these modes effectively at runtime. One common scheme, which has been implemented in HP and IBM servers, is to enforce an aggressive power cap against a group of machines – individual machines in the group are permitted to consume a large amount of power, but others will be throttled to obey the power budget. Other approaches involve distributing incoming work across machines to maximize energy efficiency.

All of these approaches can benefit from real-time models that can predict a computer's instantaneous power consumption, based on resource utilization metrics available in software. These models are different from detailed, component-specific architectural models; they are designed to be high-level and low-overhead so that they can be deployed non-intrusively. Previous work on full-system power modeling has assumed that the major consumers of dynamic power are the CPU, memory, and disk. However, the increased use of GPUs for general-purpose computing [20] introduces a new, understudied variable in the current dialog on energy efficiency.

GPUs are specialized for data-parallel computation, which includes many scientific and cryptographic applications. Three of the top five supercomputers in the world use GPUs (http://www.top500.org), and future cloud computing infrastructure is likely to be heterogeneous—including CPUs and GPUs.

While CPUs' power consumption has improved significantly in recent years, GPUs have not experienced the same aggressive energy-efficiency optimization. GPUs perform much better than CPUs at certain specialized types of computation, but their power consumption is also high. Therefore, models that exclude the GPU may significantly mispredict power consumption on important workloads, leading to decisions that waste power or even exceed power budgets for a facility.

In this research, I determined that traditional power models are highly inaccurate for GPU-intensive workloads. I then modified these models to be GPU-aware, greatly increasing the accuracy of their predictions.

2. BACKGROUND AND RELATED WORK

Several studies have modeled computers' power consumption [3, 9, 6, 7, 10, 13, 14, 15, 16, 18]. However, prior work has targeted traditional desktop and server workloads, which do not engage the GPU. Still other studies seek to model the power of CPUs alone from microarchitectural information [5, 12]. My work uses the Mantis methodology and toolset to build full-system power models [18], to which I added GPUs.

Because GPUs have a relatively short history in enterprise and scientific computing, few studies model GPU power. Of those studies, two modeled the power consumption of GPUs alone, rather than their contribution to system-level power [11, 17]. These models also use lower-level architectural information than my models, which are high-level and portable across GPUs and systems.

Bircher modeled the system-level power of a machine that included GPUs [2]. That study's goal was to build low-level models for a single, specific machine, maximizing accuracy. In addition, those models were validated using traditional graphics workloads rather than newer general-purpose and scientific GPU workloads. In contrast, my goal was to build high-level, simple models and maximize accuracy within that constraint. Furthermore, I quantify the benefits of including GPUs in the system power model, and I show which high-level counters best predict GPUs' contribution to system-level power consumption.

3. APPROACH AND UNIQUENESS

I built full-system power models for two machines. The machines have different processors and different generations GPUs.

By having multiple generations and families of hardware between the two systems, the model becomes truly generic and portable. This avoids models that are tied to highly specific/ proprietary microarchitectural features, such as the processor model or the chip layout. The generational difference also yields insight about how energy efficiency is affected by hardware trends, from advances like dynamic fan speed, number of cores, more CPU frequency ranges etc.

System	CPU	Memory	GPU
System 1 *	AMD Athlon 64 X2 4800+ 2.5 GHz (2 cores)	4 GB	nVidia 8800gt
System 2**	Intel Core i5-750 2.66 GHz (4 cores)	8 GB	nVidia GeForce GTX 285 GP

* built prior to 2008, GPU added in 2008

** built late 2008-early 2009

3.1 Mantis Approach

I extended the Mantis software to build these power models. Mantis uses the following approach:

1. **Training** (see Fig. 1)

- Plug the system under test into a power meter, which plugs into the wall and measures the AC power consumed by the system. Connect the power meter's USB cable to a separate control and measurement system to avoid perturbing the behavior of the system under test.

  The power meters I used sample the wall power at 1 Hz, with a measurement error of 1.5%.

- Run synthetic benchmarks to stress the different components of the system under test. The original Mantis training benchmarks stress the CPU, memory, and disk; I added a GPU stress component to these benchmarks.

- Collect CPU and disk utilizations for the system under test at a rate of one sample per second. Collecting this data has minimal overhead on the system under test.

- After the benchmarks have run, combine the AC power measurements and utilization data on the control and measurement system.

2. **Model fitting:**

   After the training phase, we have a vector of utilization measurements and power consumption with one entry for each second of the synthetic benchmarks' execution.

   The next step is to fit model coefficients to this calibration data. I used linear regression as I found that it yielded sufficiently accurate models to make the points described in the next section.

   After this fitting, the model can be used to predict the system's wall power by sampling the utilizations and plugging them into the linear equation obtained in the previous step.

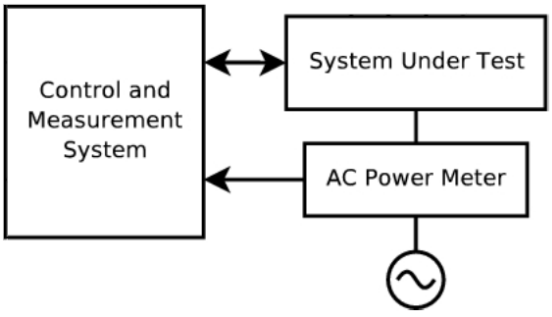

Figure 1. Instrumentation infrastructure.

3.2 Models Built

I built the following models for each of the two machines at each available CPU frequency:

1. A purely traditional model. The training data for this model included synthetic benchmarks designed to stress the CPU, memory, and disk. The predictors for this model were CPU utilization and disk utilization, and the model was fit using linear regression.

2. A variation on the first model. This model used the same predictors as the first model (CPU and disk utilization). The difference was in the training data used to fit the model. This training data included the benchmarks designed to stress CPU, memory, and disk as well as a benchmark designed to stress the GPU.

3. A series of models that used various GPU metrics from the Nvidia-SMI instrumentation suite as predictors. I evaluated their accuracy and quantified the usefulness of each GPU metric.

4. RESULTS AND CONTRIBUTIONS

4.1 Traditional Models

First, I built linear models using only CPU and disk utilizations (high-level, portable metrics) as predictors. The training data was the traditional Mantis calibration suite, consisting of CPU, memory, and disk stress benchmarks. The R^2 values for these models ranged between 0.80 and 0.85, indicating the models were correctly able to predict most variation in the power consumption for the traditional training workload.

Figure 2 shows the actual power (black) and predicted power (red) of a traditional model when applied to the SPEC JBB benchmark. SPEC JBB is a traditional CPU- and memory-intensive benchmark. The predictions follow the shape of the actual power, and the errors are no higher than 10% of the actual power. Overall, running SPEC JBB at different processor frequencies yielded mean-squared prediction errors of 6.04 to 98.28 on the older machine and 68.08 to 284.05 on the newer one. By contrast, running the nbody, FDTD3d, and binomial options workloads yielded mean squared errors (MSE) of up to 631.19 on the older machine and 1793.59 on the newer one.

Figure 3 shows the same model applied to a GPU-intensive workload (an N-body simulation). Here, the model does not predict the GPU's spike in power, and the error is approximately 50% of the measured power. This graph shows that the GPU is a significant consumer of dynamic power whose activity is largely uncorrelated to the CPU's.

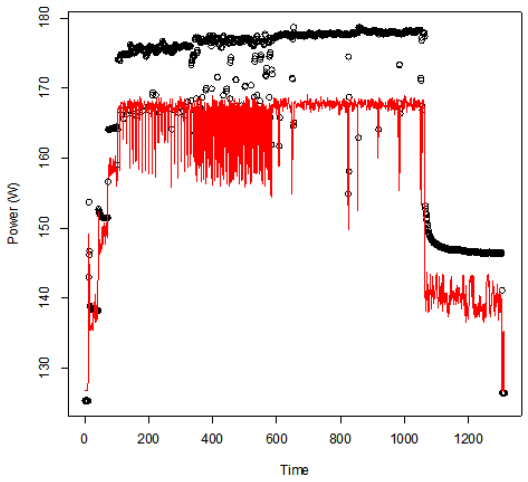

Figure 2. Measured power (black) for a CPU-intensive workload. The predicted power, using a traditional model, is shown in red.

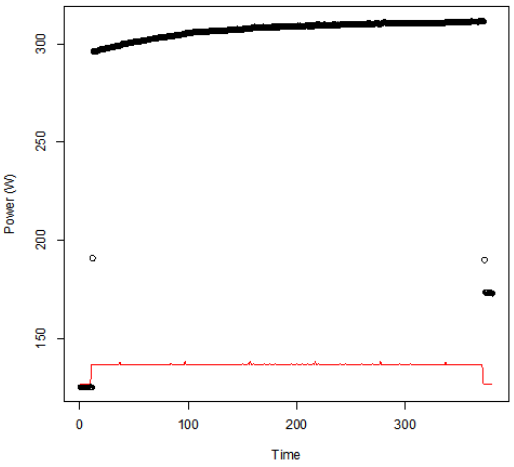

Figure 3. Measured power (black) for a GPU-intensive workload. The predicted power, using a traditional model, is shown in red.

4.2 Traditional Models with GPU Training

The next models I built were trained on workloads that included the GPU. However, they still used only CPU and disk utilization as predictors. Figure 3 and 4 show the results.

Figure 4 shows the results of this model when predicting the power of the matrix multiplication phase of the calibration program. Since this program is part of the training set, I would expect the accuracy of these predictions to be artificially high. Even so, these predictions are much less accurate than those in Figure 2.

Figure 5 shows the results of this model when predicting the power of the GPU stress program from the training suite. Here, the error is not as bad as the error from Figure 3. For these models, the R^2 range was 0.531 to 0.545, showing that the CPU and disk utilizations do not capture the activity of the GPU.

The lesson of Figures 4 and 5 is that a GPU predictor is needed in the model. Including GPU workloads in the training data, but not adding GPU predictors, forces the CPU utilization to account for all of the dynamic power variation in the workloads. This leads to an inflated model coefficient for CPU utilization, resulting in the excessive peaks and valleys shown in Figure 4. The prediction accuracy does improve for the GPU workload, but the exaggerated response to CPU utilization destroys the accuracy for CPU workloads.

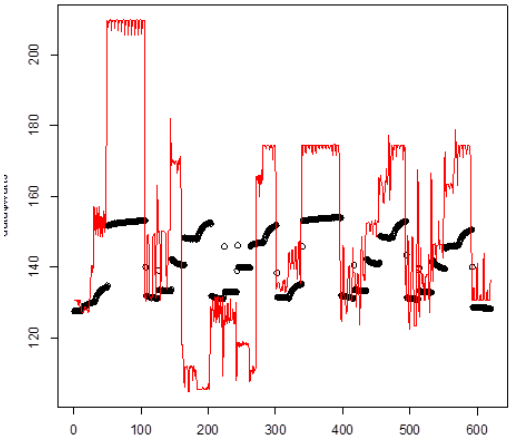

Figure 4. Measured power (black) for a CPU-intensive workload (left). The predicted power, using a traditional model trained on GPU-aware data, is shown in red.

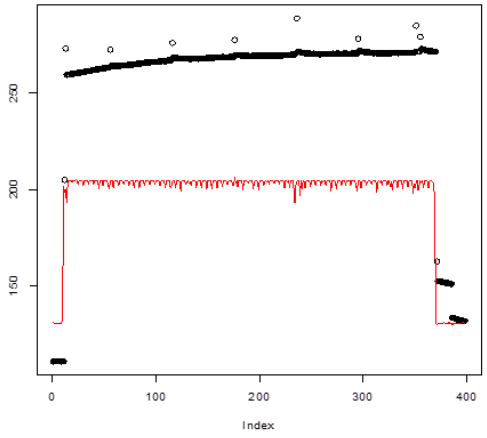

Figure 5. Measured power (black) for a a GPU-intensive workload (right). The predicted power, using a traditional model trained on GPU-aware data, is shown in red.

4.3 GPU Predictor Models

Finally, I built models that included GPU predictors. Figure 6 shows the R^2 of these models when fitted to training data that includes CPU, memory, disk, and GPU workloads. In Figure 6, the blue bars correspond to the newer test machine, and the red bars correspond to the older machine.

In Figure 6, Model A is the original model (CPU + disk utilization) using the CPU + GPU training data. All of the other models include additional predictors beyond CPU and disk:

- Model B uses GPU utilization
- Model C uses GPU memory utilization
- Model D uses GPU temperature
- Model E uses GPU fan speed
- Model F uses all of the metrics in B-E

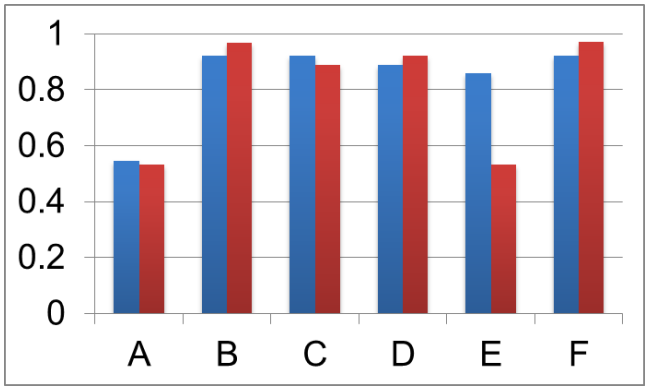

Figure 6. Comparison of GPU-aware models (R2). Models B-F use different GPU metrics. Model A is not GPU-aware.

In order to understand the individual contribution of each GPU-related predictor, I built models that included only the traditional CPU-centric predictors plus each GPU predictor in turn.

When I built these models individually, the results showed that GPU Utilization (Model B) is the single best predictor, with MSE range of 22.52-38.24 and R^2 range of 0.92-0.96. GPU Memory (Model C) and GPU Temperature (Model D) were also consistently good (R^2 range 0.88-0.92).

GPU Fan Speed (Model E) had some success, with an R^2 range of 0.53-0.85, but on the older machine, it remained constant throughout the workload and dropped out of the model because the older GPU does not have dynamic fan speed. I also tried the cube of GPU Fan Speed (since power grows cubically with fan speed), but it did not significantly improve the predictions.

Model F includes all of the GPU predictors. For this model, R^2 falls within a range of 0.922 to 0.970. The MSE also reflects a large improvement, ranging from 21.4-37.69. However, Model F barely outperforms Model B, which includes fewer predictors. To minimize the risk of overfitting, Model B is probably preferable to Model F for deployment in the wild.

4.4 Future Work

The results of this work suggest several directions for future research. First, it would be valuable to validate the conclusions of this work against a larger set of GPU benchmarks, encompassing traditional graphics workloads as well as a broader range of scientific workloads; the Parboil benchmark suite from the University of Illinois

could address the latter objective. It would also be useful to validate the benchmark against heterogeneous CPU+GPU workloads, such as the Rodinia benchmark suite from the University of Virginia.

A second avenue for future research would be to investigate lower-level predictors in order to understand how much accuracy I sacrificed by using a high-level approach. On the traditional CPU and memory side, processor-specific hardware performance counters have been used in previous power modeling studies. On the GPU side, manufacturers offer profiling toolkits that convey more detailed information than nVidia-smi. Although these lower-level predictors are more intrusive and less general, they would probably lead to more accurate models.

Finally, along similar lines, it would be worth exploring more complex model types that go beyond simple linear regression. In particular, some predictors (like fan speed and temperature) probably contribute nonlinearly to power consumption.

4.5 Conclusions

This work shows that simple GPU metrics such as GPU utilization significantly improve the accuracy of high-level power models when applied to GPU workloads. First, it shows that while traditional non-GPU-aware models clearly capture the power consumption of traditional workloads (Figure 2), these models struggle to predict the behavior of GPU workloads (Figure 3). When the traditional model trains on GPU-aware workloads, the model forces the CPU utilization to account for the GPU's behavior, causing amplified peaks and dips on CPU workloads (Figure 4) and failing to predict the GPU's full power consumption (Figure 5). Only when the model includes GPU predictors as well as the traditional CPU, memory, and disk does a new picture emerge of the true influence the GPU is having on power consumption.

ACKNOWLEDGMENTS

This research was funded by the Computing Research Association's Collaborative Research Experience for Undergraduates (CREU) and by a Sonoma State University Undergraduate Research Grant. Collaborators on this work include Prof. Suzanne Rivoire (research advisor), Benjamin Morrison and Vincent Morrow (SSU undergraduates), and Forrest Lipske (high school intern).

REFERENCES

[1] L.A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," IEEE Computer, vol. 40, no. 12, pp. 33-37, Dec. 2007.

[2] W.L. Bircher and L.K. John, "Complete System Power Estimation Using Processor Performance Events," IEEE Transactions on Computers, preprint, 10 Feb 2011.

[3] W.L. Bircher and L.K. John, "Complete System Power Estimation: A Trickle-Down Approach Based on Performance Events," in Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2007.

[4] S. Bird, "Fixing Performance Counters: Performance Monitoring Hardware for the Datacenter," in Workshop on Architectural Concerns in Large Datacenters (ACLD), 2009.

[5] G. Contreras and M. Martonosi, "Power Prediction for Intel XScale Processors Using Performance Monitoring Unit Events," in Proceedings of the International Symposium on Low-Power Electronics and Design (ISLPED), 2005.

[6] J.D. Davis, S. Rivoire, M. Goldszmidt, and E.K. Ardestani, "Accounting for Variability in Large-Scale Cluster Power Models," in Proceedings of the Exascale Evaluation and Research Techniques Workshop (EXERT), 2011.

[7] John D. Davis, Suzanne Rivoire, Moises Goldszmidt, and Ehsan K. Ardestani, No Hardware Required: Building and Validating Composable Highly Accurate OS-based Power Models, Microsoft Research Technical Report no. MSR-TR-2011-89, July 2011

[8] United States Environmental Protection Agency, "Report to Congress on Server and Datacenter Energy Efficiency," 2 Aug. 2007. Available online at: http://www.energystar.gov/index.cfm?c=prod_development.server_efficiency_study

[9] X. Fan, W.-D. Weber, and L.A. Barroso, "Power Provisioning for a Warehouse-Sized Computer," in Proceedings of the International Symposium on Computer Architecture (ISCA), 2007.

[10] T. Heath, B. Diniz, E.V. Carrera, W. Meira, Jr., and R. Bianchini, "Energy Conservation in Heterogeneous Server Clusters," in Proceedings of the Symposium on Principles and Practice of Parallel Programming (PPoPP), 2005.

[11] S. Hong and H. Kim, "An Integrated GPU Power and Performance Model," in Proceedings of the International Symposium on Computer Architecture (ISCA), 2010.

[12] C. Isci and M. Martonosi, "Runtime Power Monitoring in High-End Processors: Methodology and Empirical Data," in Proceedings of the 36th International Symposium on Microarchitecture (MICRO), 2003.

[13] R. Koller, A. Verma, and A. Neogi, "WattApp: An Application Aware Power Meter for Shared Data Centers," in Proceedings of ACM International Conference on Autonomic Computing (ICAC), 2010.

[14] A. Lewis, S. Ghosh, and N.-F. Tzeng, "Run-Time Energy Consumption Estimation Based on Workload in Server Systems," in Proceedings of the Workshop on Power-Aware Computing and Systems (HotPower), 2008.

[15] A. Lewis, J. Simon, and N.-F. Tzeng, "Chaotic Attractor Prediction for Server Run-Time Energy Consumption," in Proceedings of the International Conference on Power-Aware Computing and Systems (HotPower), 2010.

[16] T. Li and L.K. John, "Run-Time Modeling and Estimation of Operating System Power Consumption," in Proceedings of SIGMETRICS, 2003.

[17] X. Ma, M. Dong, L. Zhong, and Z. Deng, "Statistical Power Consumption and Analysis for GPU-Based Computing," in

Proceedings of the 2nd Workshop on Power Aware Computing and Systems (HotPower), 2009.

[18] S. Rivoire, P. Ranganathan, and C. Kozyrakis, “A Comparison of High-Level Full-System Power Models,” in Proceedings of the 1st Workshop on Power Aware Computing and Systems (HotPower), 2008.

[19] D.C. Snowdon et al., “Koala: A Platform for OS-Level Power Manage- ment,” in Proceedings of the ACM European Conference on Computer Systems (EuroSys), 2009.

[20] S. Suneja, E. Baron, E. de Lara, and R. Johnson, “Accelerating the Cloud with Heterogeneous Computing,” in Proceedings of the 3rd USENIX Workshop on Hot Topics in Cloud Computing, 2011.