# Including Variability in Large-Scale Cluster Power Models

John D. Davis[1], *Member, IEEE,* Suzanne Rivoire[2], *Member, IEEE,* Moises Goldszmidt[1], *Member, IEEE,* and Ehsan K. Ardestani[3], *Student Member, IEEE*

Microsoft Research - Silicon Valley Lab[1] [john.d, moises]@microsoft.com

Sonoma State University[2] suzanne.rivoire@sonoma.edu

University of CA, Santa Cruz[3] eka@soe.ucsc.edu

**Abstract**—Studying the energy efficiency of large-scale computer systems requires models of the relationship between resource utilization and power consumption. Prior work on power modeling assumes that models built for a single node will scale to larger groups of machines. However, we find that inter-node variability in homogeneous clusters leads to substantially different models for different nodes. Moreover, ignoring this variability will result in significant prediction errors when scaled to the cluster level. We report on inter-node variation for four homogeneous five-node clusters using embedded, laptop, desktop, and server processors. The variation is manifested quantitatively in the prediction error and qualitatively on the resource utilization variables (features) that are deemed relevant for the models. These results demonstrate the need to sample multiple machines in order to produce accurate cluster models.

## 1 Introduction

POWER consumption is a major concern in the design and operation of large-scale computing facilities [2]. It also presents a modeling and instrumentation challenge to researchers and infrastructure providers.

Physical instrumentation alone is not sufficient for challenges such as attributing power consumption to virtual machines, predicting how power consumption scales with the number of machines, and predicting how changes in utilization affect power consumption. These tasks require accurate models of the relationship between resource usage and power consumption. Furthermore, measurement adds significant cost to the system.

A substantial body of literature models power consumption by sampling various metrics available in software (CPU utilization, memory bandwidth, disk utilization, etc.) and fitting them to the measured system-level power consumption of a node. (Note that we use the terms *node* and *machine* interchangeably.) However, most of this previous work has built and validated models for individual nodes, with the implicit or explicit assumption that these models would extrapolate to the cluster level and beyond.

In this paper, we test that assumption by building node-level and cluster-level power models for four homogeneous clusters running MapReduce-style applications. The clusters include components from the embedded, mobile (laptop), desktop, and server processor spaces, reflecting energy-efficient server recommendations from recent research [1], [9], [13], [26] as well as traditional servers prevalent today.

Our results clearly demonstrate that single-node power models do not scale to the cluster level:

- We show that the model correlates (or model features) chosen for single-node models by a standard feature selection process vary across individual nodes in a homogeneous cluster.
- We further show that, for a given set of features, the coefficients of a fitted single-node model are highly sensitive to the particular node.

We observe that node-to-node variation is distinct from, and an order of magnitude higher than, run-to-run variation on these four clusters. Manufacturing variation among "identical" components has been documented by others [18], [21]. Our goals are two-fold: (1) document this variability at the cluster level and (2) present an approach to build cluster power models that tolerates variability.

## 2 Related Work

Previous studies model the power consumption of single nodes using different predictors and modeling techniques [3],[4],[12],[14],[23],[24],[25]. Some studies predict power consumption based only on CPU utilization [7], [19], while others use board-level measurements [16]. The modeling techniques also vary in complexity, from simple lookup-based models [22] to chaotic attraction predictors [16]. These studies all build and validate models on a single node, assuming that these models can be applied to other identically configured nodes without requiring refitting. We challenge that assumption in this work.

Other studies use different validation techniques. Li and John validate their routine-specific models on a full-system simulator [17], which again assumes no inter-node variability. Vasan et al. present power measurements from a medium-scale datacenter but only build single-node models [27]. Heath et al. model the total power of an eight-node heterogeneous cluster on a single workload that exhibits little dynamic variation; their work does not address the question of scaling the model to include additional nodes [10]. Lang and Patel model the energy, rather than the instantaneous power, of a 24-node cluster

* Manuscript submitted: 16-Sep-2011. Manuscript accepted: 18-Oct-2011. Final manuscript received: 25-Oct-2011.

Posted to the IEEE & CSDL on 11/17/2011
DOI 10.1109/L-CA.2011.27 1556-6056/11/$26.00 © 2011 Published by the IEEE Computer Society

TABLE 1
PLATFORMS FOR FULL-SYSTEM POWER MODELING
(* = Maximum memory capacity of the system)

| System Class | CPU | Memory | Disk(s) | OS, FS |
|---|---|---|---|---|
| Embedded | Intel Atom, dual-core, 1.6 GHz, 8W TDP [26] | 4 GB DDR2-800* | 1 Micron SSD | Windows Server 2008 R2, NTFS |
| Mobile | Intel Core 2 Duo, dual-core, 2.26 GHz, 25W TDP [13] | 4 GB DDR3-1066* | 1 Micron SSD | |
| Desktop | AMD Athlon, dual-core, 2.8 GHz, 65W TDP [9] | 8 GB DDR2-800 | 1 Micron SSD | |
| Server | AMD Opteron, quad-core, 2.0 GHz, 50W TDP | 32 GB DDR2-800 | 2 10K RPM SATA | |

[15]; it is unclear whether they do so by scaling the measured power consumption of a single node. Finally, Fan et al. scale a single-node, CPU utilization-based power model to a few hundred servers [8]. However, they must add a large constant offset to the predicted power, which compensates for the constant power consumption of networking equipment as well as inter-node variations in idle power. They do not separate these two components of the added offset.

Manufacturing variation in power among "identical" hardware components, which is the central challenge of this paper, has been well documented [21], [18]. McCullough et al. also show that power variation among cores in a multicore CPU can harm the accuracy of power models. However, they do not examine the question of how to make models tolerant of variation, and they do not look beyond a single node [18].

## 3 SYSTEM OVERVIEW

We build models for four homogeneous five-node clusters running data-intensive, MapReduce-style applications. In this section, we describe the hardware platforms, the software infrastructure, and the workloads used to build large-scale power models.

### 3.1 Hardware Infrastructure

Our systems have different CPU dynamic voltage and frequency scaling (DVFS) capabilities, which affects the resulting power models. Table 1 lists the features of these systems. Starting at the low end, the Atom N330 processor does not provide DVFS at all. This cluster also has the smallest dynamic power range, on the order of 15W over the entire cluster. On the other hand, the mobile- and desktop-processor-based systems both use DVFS. For these two systems, the two cores on a single node report the same operating frequency 99.8% of the time for our workloads. Finally, the server-class system has the ability to have the cores operate in different p-states (frequency), and can transition the system into the C1 idle state when all processors are idle. For our workloads, the frequencies of the cores on a single server node differed up to 12% of the time.

Each machine reads its own power measurements over a USB port. The power meters have an error of 1.5%. We verified the meter calibration, but we leave the explicit extraction of meter error for future work.

### 3.2 Software Infrastructure

Each system runs Windows Server 2008 R2, which provides a standardized OS-level performance counter interface. We measure a wide range of Event Tracing for Windows (ETW) performance counters provided by the OS. For each machine, we collect metrics, at 1 Hz, relating to the processor, memory, physical disk, process, job object, file system cache, and network interfaces [20]. Overall, we collect approximately 250 counters per node. Statistically redundant counters are removed through a systematic feature selection process, described in Section 4.1. We also verified that the data collection process does not interfere with program behavior or power consumption. Table 2 lists the final subset of performance counters used by the various cluster models (6-8 counters per model); see [6] for more details on model feature selection.

We ran an assortment of distributed workloads using the Dryad and DryadLINQ application framework [11]. These workloads are diverse; some are CPU-intensive, while others are dominated by disk and network. We run a single instance of each application at a time, five times per cluster to allow each node to act as the job scheduler, which provides diversity in the work done even for the same application. One machine acts as the job manager, and the other four machines compute the tasks from the task graph. The workloads used are described below:

- **Sort**: sorts 4GB of data with 100-byte records. The data is separated into 20 partitions, distributed randomly across the cluster. All of the data must first be read from disk and ultimately transferred back to disk on a single machine, so this workload has high disk and network utilization.
- **PageRank**: runs a graph-based page ranking algorithm over the billion-page ClueWeb09 dataset [5], spread over 80 partitions on a cluster. It is a 3-step job in which output partitions from one step are fed as inputs to the next step. Thus, PageRank has high network utilization.

TABLE 2
ETW PREDICTORS USED IN CLUSTER POWER MODELS

| Category | Performance counter | Ctr. ID |
|---|---|---|
| Memory (Mem) | Page Faults/sec | 18 |
| | Cache Faults/sec | 24 |
| | Pages/sec | 26 |
| | Pool Nonpaged Allocs | 34 |
| Physical Disk (PD) | Disk Total Disk Time % | 54 |
| | Disk Total Disk Bytes/sec | 66 |
| Process (Proc) | Total IO Data Bytes/sec | 99 |
| Processor (uP) | Total Processor Time % | 102 |
| File System Cache (FSC) | Data Map Pins/sec | 121 |
| | Pin Reads/sec | 122 |
| | Copy Reads/sec | 126 |
| | Fast Reads Not Possible/sec | 139 |
| | Lazy Write Flushes/sec | 140 |
| Job Object Details (JOD) | Total Page File Bytes Peak | 167 |
| Proc. Perf. (MHz) | Processor_0 Frequency | 209 |

- **Prime**: checks primeness of approximately 1,000,000 numbers over 5 partitions in a cluster. It has high CPU usage but little network traffic.
- **WordCount**: reads through 50 MB text files on 5 partitions in a cluster and tallies the occurrences of each word. It has little network traffic.

## 4 Model and Machine Variability

In this section, we demonstrate the variation in feature selection and model coefficients across "homogeneous" nodes. We also show the impact on accuracy of scaling up a single-node model and compare it to model creation based on a sample of multiple nodes.

### 4.1 Model Creation and Feature Variability

We evaluated four classes of power models: linear, piecewise linear, quadratic, and switching. For brevity's sake, we present only the best linear models for each cluster. The overall predicted cluster power is the sum of the single-machine models built using the metrics from each machine (1).

$$Power_{cluster} = \sum_{i=1}^{n} Power_{machine_i} \quad (1)$$

The challenge was to produce a single-node model that provides the lowest root-mean-squared error across all workloads on the cluster. We report this error as a percentage of the cluster's *dynamic* power range; we refer to this metric as dynamic range error (DRE). Equation (2) gives the formula for DRE.

$$Error\ (DRE) = \frac{\sqrt{Mean\ Square\ error}}{Max\ Power_{Cluster} - Min\ Power_{Cluster}} \quad (2)$$

Equations (3) through (6) below show the final features for each cluster-specific model. Power is denoted by $y$, and $x_i$ denotes the features measured on each machine. We collect data of the form <$y$, $x_1$,…,$x_n$>, and we fit functions $\hat{f} = (x_1, \ldots x_n)$ so that $\hat{f}()$ approximates $y$, minimizing some loss function. The numeric subscripts refer to the counter IDs in Table 2.

$$P_{Server} = \hat{f}(x_{26}, x_{54}, x_{66}, x_{102}, x_{121}, x_{122}, x_{167}, x_{209}) \quad (3)$$

$$P_{Desktop} = \hat{f}(x_{18}, x_{26}, x_{99}, x_{102}, x_{167}, x_{209}) \quad (4)$$

$$P_{Mobile} = \hat{f}(x_{24}, x_{34}, x_{54}, x_{102}, x_{122}, x_{167}, x_{209}) \quad (5)$$

$$P_{Emb.} = \hat{f}(x_{24}, x_{34}, x_{66}, x_{102}, x_{121}, x_{126}, x_{139}, x_{140}, x_{167}) \quad (6)$$

Our previous work [6] provides details on which performance counters are significant for each individual node and each workload for all the benchmarks. The feature selection heuristic is presented for the clusters based on the sum of an individual node's significant features. We found there to be considerable variation in features selected by a model across different nodes in the cluster.

### 4.2 Coefficient Variability and Overall Accuracy

With the model features selected, we built single-node power models for each cluster and used two different methods to scale these models to predict cluster power. We estimate the cluster-level power models' error using five-fold cross validation using training data from all the workloads. The training and test runs vary based on what input data runs on which node in the cluster.

The first strawman method used to predict cluster power was to build a model to predict the power of a single node, and then simply multiply this predicted power by the number of nodes in the cluster. Unsurprisingly, this method was highly inaccurate, yielding worst-case dynamic range errors of up to 150%.

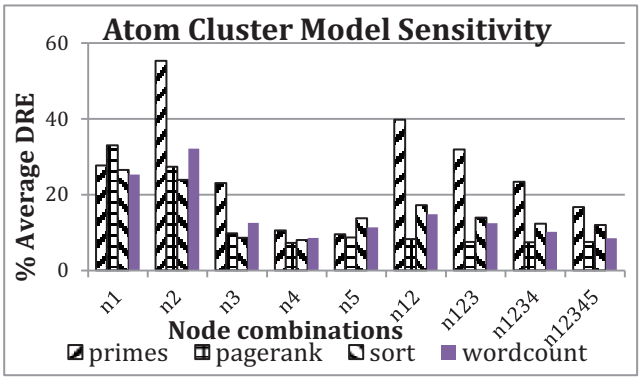


Fig. 1. Sensitivity to the machine(s) used to train cluster models. Column labels identify the node(s) used in training.

The second method collects performance counter data from all nodes and applies the single-node model to each node in turn, summing the predictions. Fig. 1 shows the results of this method. For each cluster, columns n1 through n5 show the dynamic range error when the cluster models are trained using data from only one node (each of nodes 1-5) and then applied to all nodes. The remaining columns show models trained using data from subsets of the five nodes (i.e. n12 is a model trained on nodes 1 and 2 and applied to the entire cluster). Using data from multiple machines is far superior to simply scaling a single node's power, decreasing the worst-case error to only ~50% for the Embedded cluster compared to ~150% when multiplying a single node's predicted power by N, the number of machines modeled in the cluster.

As Fig. 1 shows, the machine power model trained using a particular node was sometimes a good proxy for cluster power model coefficients, while in other cases it was not. In general, as we added more machine data from different nodes of the cluster to train the model and determine the feature coefficients, the accuracy of the linear model improved, reducing worst-case error from ~50% down to less than 20% for the Embedded cluster and 10% for the other clusters. Using quadratic models, the worst-case DRE went down to 12% for the Embedded cluster running WordCount or 1% absolute median error. PageRank was the worst-case absolute median error was 5.7% on the Desktop cluster (9% DRE). All other absolute median errors were 3.7% or less.

For large-scale data centers, it is impractical to train the model with all the machines in the data center. In our prior work [6], we formally derive the number of machines that must be sampled to meet a given error bound based on the measured power difference across machines.

#### *Application inter-run variation*

We also compared the run-to-run variation in idle power of the individual nodes to the machine-to-machine variation in idle power in the cluster. The inter-run idle power range for a single node was as much as an order of magnitude smaller than, and never larger than the cluster

idle power range. These results, shown in our prior work [6], demonstrate that multiple application run measurements on the same node are not sufficient to capture the inter-node variability that we have observed on the server cluster.

### *Meter error vs. measured power ranges*

The Watts-Up Pro meter error is reported as 1.5%. When looking at the idle, average, and maximum power ranges across all the clusters and benchmarks, only the measured ranges for the Opteron cluster is less than ±1.5% of the possible meter error. All other clusters report measured power ranges greater than the meter error for at least one application on the cluster. The error ranges have been omitted for brevity. Simply using measurement error does not capture machine variability for all the clusters.

## 5 Conclusions

Previous work assumes that it is sufficient to build and then scale a single-node power model for each system class of interest. For high-fidelity cluster power models, our results show that the choice of model predictors will vary from node to node. Furthermore, even for a given set of predictors, inter-node variability will result in different model coefficients when models are fit using data from different individual nodes. As one would expect, these variations in single-node models result in larger errors than using multiple nodes to train the models for predicting cluster-level power consumption.

We also observed greater inter-node measured power variation than run-to-run variation on a single node, requiring models based on a sample from the population of machines. Although not presented here, the number of machines to sample is independent of the machine population size and given reasonable parameters is on the order of a single rack of machines or less [6].

Finally, the combination of the portable (across different machine types) ETW framework, feature selection heuristic, sampling bounds, and standard statistical methods provides a methodology that can be easily applied to new clusters composed of different systems and/or new workloads to generate high-fidelity full-system cluster power models.

## References


[1] D. Andersen et al., "FAWN: A Fast Array of Wimpy Nodes," *Proc. ACM Symp. Operating Systems Principles*, pp. 1-14, Oct. 2009.

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

[3] W.L. Bircher and L.K. John, "Complete System Power Estimation: A Trickle-Down Approach Based on Performance Events," *Proc. IEEE Int'l Symp. Performance Analysis of Systems and Software*, Apr. 2007.

[4] J. Choi et al., "Profiling, Prediction, and Capping of Power Consumption in Consolidated Environments," *Proc. IEEE Int'l Symp. Modeling, Analysis and Simulation of Computers and Telecommunications Systems*, pp. 1-10, Sep. 2008.

[5] ClueWeb09 dataset, available at: http://boston.lti.cs.cmu.edu/Data/clueweb09/

[6] J.D. Davis et al., "Accounting for Variability in Large-Scale Cluster Power Models," *Proc. Workshop Exascale Evaluation and Research Techniques*, Mar. 2011.

[7] G. Dhiman, K. Mihic, and T. Rosing, "A System for Online Power Prediction in Virtualized Environments Using Gaussian Mixture Models," *Proc. ACM Design Automation Conference*, pp. 807-812, Jun. 2010.

[8] X. Fan, W.-D. Weber, and L.A. Barroso, "Power Provisioning for a Warehouse-Sized Computer," *Proc. ACM Int'l Symp. Computer Architecture*, pp. 13-23, Jun. 2007.

[9] J. Hamilton, "CEMS: Low-Cost, Low-Power Servers for Internet-Scale Services," *Proc. Conf. Innovative Data Systems Research*, Jan. 2009.

[10] T. Heath et al., "Energy Conservation in Heterogeneous Server Clusters," *Proc. ACM Symp. Principles and Practice of Parallel Programming*, pp. 186-195, Jun. 2005.

[11] M. Isard et al., "Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks," *Proc. European Conf. Computer Systems*, pp. 59-72, Mar. 2007.

[12] A. Kansal et al., "Virtual Machine Power Monitoring and Provisioning," *Proc. ACM Symp. Cloud Computing*, pp. 39-50, Jun. 2010.

[13] L. Keys, S. Rivoire, and J.D. Davis, "The Search for Energy-Efficient Building Blocks for the Data Center," *Proc. Workshop Energy-Efficient Design*, Jun. 2010.

[14] R. Koller, A. Verma, and A. Neogi, "WattApp: An Application Aware Power Meter for Shared Data Centers," *Proc. ACM Int'l Conf. Autonomic Computing*, pp. 31-40, Jun. 2010.

[15] W. Lang and J.M. Patel, "Energy Management for MapReduce Clusters," *Proc. VLDB Endowment*, vol. 3, no. 1-2, pp. 129-139, Sep. 2010.

[16] A. Lewis, J. Simon, and N.-F. Tzeng, "Chaotic Attractor Prediction for Server Run-Time Energy Consumption," *Proc. Intl. Conf. Power-Aware Computing and Systems*, pp. 1-16, Oct. 2010.

[17] T. Li and L.K. John, "Run-Time Modeling and Estimation of Operating System Power Consumption," *Proc. ACM Int'l. Conf. Measurement and Modeling of Computer Systems*, pp. 160-171, Jun. 2003.

[18] J. McCullough, et al., "Evaluating the Effectiveness of Model-Based Power Characterization," *In Proceedings of the USENIX Annual Technical Conference,* Portland, June 2011.

[19] D. Meisner and T.F. Wenisch, "Peak Power Modeling for Data Center Servers with Switched-Mode Power Supplies," *Proc. ACM/IEEE Int'l Symp. Low-Power Electronics and Design*, pp. 319-324, Aug. 2010.

[20] Microsoft, "Windows 2000 Resource Kit Performance Counters, Counters by Object," available at: http://msdn.microsoft.com/en-us/library/ms803998.aspx

[21] K. Rajamani, et al., "Power Management for Computer Systems and Datacenters," *Proceedings of the 13th International Symposium on Low-Power Electronics and Design*, Aug. 2008.

[22] P. Ranganathan and P. Leech, "Simulating Complex Enterprise Workloads Using Utilization Traces," *Proc. Workshop Computer Architecture Evaluation Using Commercial Workloads*, Feb. 2007.

[23] S. Rivoire et al., "A Comparison of High-Level Full-System Power Models," *Proc. Workshop Power-Aware Computing and Systems*, Dec. 2008.

[24] K. Singh, M. Bhadauria, and S.A. McKee, "Real Time Power Estimation and Thread Scheduling Via Performance Counters," *Proc. Workshop on Design, Architecture, and Simulation of Chip Multi-Processors*, Nov. 2008.

[25] D.C. Snowdon et al., "Koala: A Platform for OS-Level Power Management," *Proc. ACM European Conf. on Computer Systems*, Apr. 2009

[26] A.S. Szalay et al., "Low-Power Amdahl-Balanced Blades for Data-Intensive Computing," *Proc. Workshop Power-Aware Computing and Systems*, Oct. 2009.

[27] A. Vasan et al., "Worth Their Watts? An Empirical Study of Datacenter Servers," *Proc. IEEE Int'l Symp. High Performance Computer Architecture*, pp. 1-10, Jan. 2010.