Leakage and Temperature Aware Server Control for Improving Energy Efficiency in Data Centers

Marina Zapater*, José L. Ayala†, José M. Moya‡, Kalyan Vaidyanathan§, Kenny Gross§, Ayse K.Coskun¶

*CEI Campus Moncloa UCM-UPM, Madrid 28040, Spain, marina@die.upm.es
†DACYA Universidad Complutense de Madrid, Madrid 28040, Spain, jayala@fdi.ucm.es
‡Electronic Engineering Dept., Universidad Politécnica de Madrid, Madrid 28040, Spain, josem@die.upm.es
§Oracle Physical Sciences Research Center, San Diego, CA 92121, {kalyan.vaidyanathan, kenny.gross}@oracle.com
¶ECE Department, Boston University, Boston, MA 02215, acoskun@bu.edu

Abstract—Reducing the energy consumption for computation and cooling in servers is a major challenge considering the data center energy costs today. To ensure energy-efficient operation of servers in data centers, the relationship among computational power, temperature, leakage, and cooling power needs to be analyzed. By means of an innovative setup that enables monitoring and controlling the computing and cooling power consumption separately on a commercial enterprise server, this paper studies temperature-leakage-energy tradeoffs, obtaining an empirical model for the leakage component. Using this model, we design a controller that continuously seeks and settles at the optimal fan speed to minimize the energy consumption for a given workload. We run a customized dynamic load-synthesis tool to stress the system. Our proposed cooling controller achieves up to 9% energy savings and 30W reduction in peak power in comparison to the default cooling control scheme.

I. INTRODUCTION

Energy-related costs are major contributors to the total cost of ownership in data centers today. The increase in energy consumed is due to both rising computational demands and the cooling costs for ensuring reliable operation of computer chips. Data center electricity represents 1.3% of all the electricity use in the world, and 2% in the US, yielding 250 billion kWh consumption per year worldwide [1].

In tandem with the technology scaling to 45nm and beyond, leakage has become an important component of the overall power consumption in computers. Prior work analyzing the effect of leakage on servers [2] highlights the limited usefulness of reducing cooling power by allowing temperature increases in the data center. Leakage power is exponentially dependent on temperature and higher temperatures could potentially lead to higher power consumption. Studies on leakage-temperature tradeoffs need to consider the power dynamics of individual servers under variable fan speeds. As fan power is a cubic function of fan speed, solutions based on over-provisioning of cold air into the servers can easily lead to energy inefficiency. On the other hand, using low fan speeds can increase temperature and leakage power.

In this paper, we present a leakage and temperature-aware server control mechanism that improves energy efficiency of enterprise servers in data centers. Our main contributions are as follows:

- The experimental methodology described in this paper allows to isolate, accurately measure, and control the power consumption associated with the fan speed by separating the power supply of the fans from the server. Such a setup has not been previously used for detailed temperature, leakage and dynamic energy characterization.
- We demonstrate the leakage-temperature tradeoffs in a real server as measured by the Continuous System Telemetry Harness [3]. By splitting the contribution of cooling power from that of leakage and dynamic system power, we derive models for the leakage as a function of the CPU temperature.
- We design a temperature and leakage-aware control policy to dynamically select the fan speed that minimizes the server energy and peak power consumption. Our controller uses a lookup table (LUT), which is generated based on the leakage and fan power analysis. The LUT is addressed by the workload utilization level at runtime and outputs the best fan speed for a given load. Our controller reduces energy consumption by up to 9% for a set of test workloads.

II. RELATED WORK

Related work on fan and cooling control strategies is based on the experimental observation of the over-provisioning of air-flow and over-cooling of servers. Xuefei et al. propose a thermal model-based real-time fan controller to reduce the energy consumption in CPU fans [4]. Other recent methods (e.g., [5]) use dynamic voltage frequency scaling (DVFS) together with the control of the fan speed in an energy-aware fashion. As opposed to our work, these models have not been tested on a real enterprise server.

A similar technique to ours is proposed by Wang et al. [6]. Their work provides optimal fan speed control for thermal management of servers and tackles the problem of over-cooling. However, they disregard the leakage contributions and its effects on power consumption. To the best of our knowledge, our work is the first to directly experiment with the leakage and temperature tradeoffs on enterprise servers, jointly addressing fan control and leakage power reduction.

III. EXPERIMENTAL METHODOLOGY

All the experiments proposed in this paper are performed on a presently shipping enterprise server with two SPARC T3 CPUs in 2 sockets, 32 8GB memory DIMMs and 2 SD hard drives. Each CPU has 16 cores and each core has 8 hardware threads that provide a total of 256 hardware threads. 6 fans, distributed in 3 rows of 2 fans, are located in the front part of the machine, driving air into the server. Airflow first goes through the DIMMs before reaching the CPUs. The power
supply units (PSUs) and hard disks are located on one side of the server, without interfering in the airflow.

To enable customized dynamic cooling control on the server and for quantifying the leakage power component, we first characterize the fans by verifying their speed with highly accurate vibration sensors and obtaining their power consumption values at each RPM setting. We then use independent power supplies (Agilent E3644A) to directly control each pair of fans. The power supplies are connected via RS-232 to a Data Logging and Control PC (DLC-PC), so that the fan speed can be adjusted by software scripts. The DLC-PC is also responsible for collecting runtime dynamics through the CSTM [3] running on the service processor of the enterprise server. For our experiments, the data collected through CSTM are: (i) 4 CPU temperature values (2 thermal sensors per die); (ii) 32 memory temperature values (1 per DIMM); (iii) per-core voltage and current values; and, (iv) power consumed by the whole system. These data are polled every 10 seconds, providing sufficient visibility into the runtime power and thermal behavior. CSTM runs as a part of the existing system software stack; therefore, the sensor data processing does not introduce additional overhead.

We explore all ranges of utilization scenarios by means of LoadGen, a customized dynamic load-synthesis tool that: (i) uses a core algorithm that maximally stuffs the instruction pipes of the multi-threaded CPUs so that the highest theoretically possible gate switching occurs in the chips; and (ii) allows customized dynamic profiles that can meet any desired utilization level by duty-cycling between 100% and idle at a fine granularity. When running LoadGen, the system is guaranteed to maintain the given CPU utilization and the workload is evenly spread among the cores. Using deterministic load profiles enables easy characterization of the system behavior.

IV. LEAKAGE AND TEMPERATURE TRADEOFFS

Leakage-temperature tradeoffs can have a high impact on the energy efficiency of enterprise servers. To evaluate them, we first perform a series of experiments. All experiments take place under the same conditions as follows: (i) the server is in an isolated environment at an ambient temperature of 24°C; (ii) the machine always starts execution from a cold state that has been previously forced by at least 10 minutes of idle execution with fans rotating at 3600RPM; (iii) at the beginning of the execution (i.e., t = 0), fan speed is set to the appropriate value, and the machine is idle for another 5 minutes to allow temperature stabilization; (iv) the last 10 minutes of the experiments are always conducted with the CPUs idle, to let temperature drop to a steady state. These conditions are selected so that experiments reflect realistic working conditions and isolate the thermal-energy issues that we want to study.

We first perform experiments to gather data at varying utilization levels and fan speeds. Data are used to derive the various contributors to the server power consumption. Once all contributions are isolated, we can evaluate the leakage-temperature tradeoffs in the server.

Exploring System Dynamics:

In order to explore the system dynamics, we run LoadGen for 30 minutes at various utilization levels (10%, 25%, 40%, 50%, 60%, 75%, 90% and 100%) with different fan speeds (1800RPM, 2400RPM, 3000RPM, 3600RPM and 4200RPM). In this case, we set the same fan speed for all three pairs of fans. Figure 1(a) shows the CPU0 temperature under 100% utilization and all the fan speeds. These experiments show interesting results for both the transient and the steady state. We observe significantly different time constants depending on the fan speed. For 1800RPM steady state is reached after 15 minutes of execution, whereas for the 4200RPM case, steady state is achieved after only 5 minutes. The magnitude of the thermal time constant is important for designing control mechanisms. The lower the fan speed, the slower the temperature reaction, which leaves more time for control decisions. However, temperature reacts much faster for the 4200RPM case, so a controller should be able to adapt faster. In addition, the considerable change in thermal time constants indicate that thermal models/predictors based on chip thermal modeling would need to take fan speeds into account to ensure accuracy in real-life settings.

Figure 1(b) shows the temperature at different workload utilization levels using a fan speed of 1800RPM. Thermal oscillations occur as LoadGen uses PWM to achieve a desired level of utilization. This plot shows the two transient temperature trends: a fast trend that raises the CPU temperature by 5°C to 8°C in less than 30 seconds due to workload changes (from idle to high utilization), and the slow temperature increase taking up to 15 minutes due to the time constants.

Leakage Model Fitting:

The results discussed above are key to analyze the different contributors to the server power consumption and to derive an empirical leakage model as in Eqn. (1). In our setup, we can separately measure and control the fan power $P_{fan}$. By monitoring the power sensors, we are also able to measure the sum of leakage power $P_{leak}$ and active power $P_{active}$ of the server. As LoadGen stresses the system at different levels of utilization ($U$) using the same workload, we can describe active power as a function of $U$. Leakage has an exponential dependence on temperature $T$ as shown in Eqn. (2), where

![Average CPU0 Temperature with Duty cycle 100%](image)

![Average CPU0 Temperature with fan speed 1800RPM](image)
Fan + Leakage

Leakage Power

70%

60%

90%

75%

65%

75%

100%

k

is a constant. Using all the measurements of power and temperature is never higher than 70 degrees consumption. Note that for all the optimum points, average an optimum fan speed that can be used to minimize the energy duration to test our controllers: (i) Test-1 ramps up and down the leakage power, and their sum for 100% utilization. The sum across all utilization values. Figure 2(a) shows the fan power, of 98%. This fitting gives an analytical model that is valid for 1 minute after each RPM update. This 1-minute value is a tradeoff between the maximum number of fan changes allowed during the execution of a highly variable workload and the maximum temperature overshoot we want to tolerate in our system. Note that 1-minute is a safe choice for our system considering the large thermal time constants.

The Bang-Bang Controller:

To determine the optimum fan speed setting that achieves the minimum energy consumption at runtime, we propose a LUT-based controller. This controller is installed in the DLC-PC, which periodically monitors the load utilization through sar and mpstat utilities. Based on the LUT output, DLC-PC then sets the fan speed to the appropriate value by increasing or decreasing the current of the power supplies. Utilization is polled every second to be able to respond to sudden utilization spikes. Polling the utilization does not introduce any noticeable overhead on the CPUs. The controller makes decisions based on changes in the load utilization rather than reacting to temperature changes, which allows the system to proactively set the optimum fan speed before a thermal event occurs.

In order to ensure the stability of the controller and to prevent fan reliability issues in the case of unstable workloads, we set a maximum frequency for the fan speed changes. We allow the controller to react fast (i.e., change fan speed as soon as a spike is detected); however, we do not allow RPM changes for 1 minute after each RPM update. This 1-minute value is a tradeoff between the maximum number of fan changes allowed during the execution of a highly variable workload and the maximum temperature overshoot we want to tolerate in our system. Note that 1-minute is a safe choice for our system considering the large thermal time constants.

The LUT-Based Controller:

The bang-bang controller only tracks the temperature using the CSTH and tries to maintain the temperature in between two desirable temperature values, 65°C-75°C, without using load utilization information. Our bang-bang controller has 5 different actions: (i) if maximum temperature $T_{max}$ goes below 60°C, fan speed is set to 1800RPM (lowest); (ii) if $T_{max}$ is in between 60°C to 65°C, fan speed is lowered by 600RPM; (iii) if $T_{max}$ is between 65 to 75 degrees, no action is taken; (iv) if $T_{max}$ is above 75°C, fan speed is increased by 600RPM; and, (v) if $T_{max}$ is above 80°C, fan speed is increased to 4200RPM. Smaller target temperature ranges (e.g., 70°C-75°C) increase fan speed change frequency whereas larger ranges (e.g., 60°C-75°C) create higher temperature overshoots and undershoots, which can lead to higher fan speeds and larger thermal cycles. The threshold values are experimentally chosen to optimize this tradeoff, ensuring the stability of the controller while keeping temperature in a range that ensures high reliability. As the time between two consecutive actions of the controller is longer than the time it takes for the temperature values to cross thresholds, no additional maximum frequency change control (as in the LUT controller) is required.

Discussion of Controller Properties:

Table I summarizes the results for all the controllers for
The bang-bang controller addresses the over-cooling in the baseline case by letting the temperature rise but keeping it low with a fan speed of 3300RPM. Controllers for Test-3. As expected, the default fan controller keeps temperature very low with a fan speed of 3300RPM. The bang-bang controller is similar to existing fan controllers in commercial servers but it allows higher temperatures. As a result, bang-bang controller generates temperature spikes and higher oscillations. The LUT controller changes fan speed according to utilization to minimize power. Even though it does not monitor temperature, the runtime temperature values are lower and more steady, so leakage is always kept low. In this test, LUT controller only needs to change the RPM between two different fan speeds because the machine is in a colder environment compared to the ambient of a data center.

Table I. Summary of controller properties

<table>
<thead>
<tr>
<th>Test</th>
<th>Control scheme</th>
<th>Energy (kWh)</th>
<th>Net Savings (kWh)</th>
<th>Peak Pwr (W)</th>
<th>Max. Temp (°C)</th>
<th>#fan change</th>
<th>Avg RPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Default</td>
<td>0.6695</td>
<td>0.6160</td>
<td>710</td>
<td>720</td>
<td>6</td>
<td>3300</td>
</tr>
<tr>
<td></td>
<td>Bang</td>
<td>0.6570</td>
<td>0.6284</td>
<td>715</td>
<td>720</td>
<td>6</td>
<td>2089</td>
</tr>
<tr>
<td></td>
<td>LUT</td>
<td>0.6556</td>
<td>0.6071</td>
<td>705</td>
<td>720</td>
<td>6</td>
<td>2117</td>
</tr>
<tr>
<td>2</td>
<td>Default</td>
<td>0.6857</td>
<td>0.6226</td>
<td>722</td>
<td>720</td>
<td>6</td>
<td>2173</td>
</tr>
<tr>
<td></td>
<td>Bang</td>
<td>0.6856</td>
<td>0.6226</td>
<td>732</td>
<td>722</td>
<td>10</td>
<td>2173</td>
</tr>
<tr>
<td></td>
<td>LUT</td>
<td>0.6685</td>
<td>0.6226</td>
<td>705</td>
<td>722</td>
<td>8</td>
<td>2181</td>
</tr>
<tr>
<td>3</td>
<td>Default</td>
<td>0.6284</td>
<td>0.6160</td>
<td>720</td>
<td>720</td>
<td>6</td>
<td>3300</td>
</tr>
<tr>
<td></td>
<td>Bang</td>
<td>0.6253</td>
<td>0.6101</td>
<td>722</td>
<td>722</td>
<td>10</td>
<td>2042</td>
</tr>
<tr>
<td></td>
<td>LUT</td>
<td>0.6262</td>
<td>0.6101</td>
<td>710</td>
<td>722</td>
<td>12</td>
<td>2161</td>
</tr>
<tr>
<td>4</td>
<td>Default</td>
<td>0.6160</td>
<td>0.6160</td>
<td>720</td>
<td>720</td>
<td>6</td>
<td>3300</td>
</tr>
<tr>
<td></td>
<td>Bang</td>
<td>0.6101</td>
<td>0.6101</td>
<td>722</td>
<td>722</td>
<td>10</td>
<td>1936</td>
</tr>
<tr>
<td></td>
<td>LUT</td>
<td>0.6071</td>
<td>0.6101</td>
<td>710</td>
<td>722</td>
<td>12</td>
<td>1968</td>
</tr>
</tbody>
</table>

**Fig. 3.** Temperature sensor readings in Test-3 for the three different controllers.

Reducing the energy consumption of enterprise servers in data centers continues to be a major challenge. As technology nodes shrink, leakage becomes an important contributor to the overall power consumption. This paper has presented an experimental methodology to explore the effect of leakage-temperature tradeoffs on the energy efficiency of enterprise servers. By utilizing a stress test, we derived an analytical model for leakage power and computed the optimum fan speeds for different utilization values. Based on our analysis, we implemented a LUT-based cooling controller that adjusts the fan speed of the system to the optimum value during runtime. Our controller provides energy savings of up to 9% and decreases the peak power by 30W. Our technique can be extended to real-life workloads by using a larger set of performance counters to characterize runtime dynamics and applying statistical analysis to derive energy-performance models for different classes of applications.

**ACKNOWLEDGMENT**

This research has been partially supported by a PICATA predoctoral fellowship of the Moncloa Campus of International Excellence (UCM-UPM), the Spanish Ministry of Economy and Competitivity under research grant TEC2012-33892, and by Oracle, Inc.

**REFERENCES**


