Proceedings of the ASME/Pacific Rim Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Systems, MEMS, and NEMS
InterPACK '09
July 19-23, 2009, San Francisco, USA

IPACK2009-89074

OPTIMAL FAN SPEED CONTROL FOR THERMAL MANAGEMENT OF SERVERS

Zhikui Wang, Cullen Bash, Niraj Tolia, Manish Marwah, Xiaoyun Zhu[†], Parthasarathy Ranganathan

Hewlett-Packard Laboratories
1501 Page Mell Road, MS 1183 Palo Alto, California 94304-1126
Email: firstname.lastname@hp.com
[†]VMware
3401 Hillview Avenue, Palo Alto, California 94304
Email: xzhu@vmware.com

ABSTRACT

Improving the cooling efficiency of servers has become an essential requirement in data centers today as the power used to cool the servers has become an increasingly large component of the total power consumption. Additionally, fan speed control has emerged in recent years as a critical part of system thermal architecture. However, the state of the art in server fan control often results in over provisioning of air flow that leads to high fan power consumption. It can be exacerbated in server architectures that share cooling resources among server components, where single hot spot can often drive the operation of a multiplicity of fans. To address this problem, this paper presents a novel multi-input multi-output (MIMO) fan controller that utilizes thermal models developed from first-principles to manipulate the operation of fans. The controller tunes the speeds of individual fans proactively based on prediction of the sever temperatures. Experimental results show that, with fans controlled by the optimal controller, over-provisioning of cooling air is eliminated, temperatures are more tightly controlled and fan energy consumption is reduced by up to 20% compared to that with a zone-based feedback controller.

1 INTRODUCTION

Power consumption is a critical issue in the design and operation of enterprise servers and data centers today. For 2006, the Environmental Protection Agency (EPA) reported that 60 billion kWh, or 1.5% of the total U.S.A. electricity consumption, was used to power data centers [1]. This is expected to rise to 100 billion kWh by 2012. In response to this problem, there have been many studies on server and cluster power management. However, server power is only one component of the total power consumed by a data center. The other significant component is power consumed by cooling equipment (e.g., fans, computer room air conditioners). Several studies [2, 3] have shown that every 1W of power used to operate a server often requires an *additional* 0.5-1W of power, needed by the cooling equipment, to extract the heat at the data center level. The same trends are applicable at the individual server level. In particular, with increasingly dense compute infrastructures, such as blade servers, and more powerful processors, the server fans can often consume a significant amount of power. Peak power usage by fans in certain blade servers can be as high as 2000W, comprising 23% of the typical system power. While a few studies have examined cooling power, they have mainly examined data center level issues [4, 5, 6, 7, 3], the state of the art in server fan control often results in over-provisioning of air flow that can lead to increased energy consumption.

In recent years, there has been a rapid growth in the use of blade servers in data centers [8]. Commercially available blade systems include HP C-Class blades, IBM BladeCenter, and Dell PowerEdge blade servers. A survey of 166 data center opera-

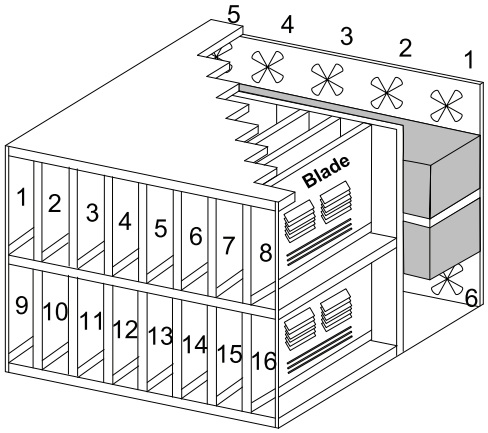


Figure 1. Enclosure Design

tors [9] showed that 76% of operators were using blade servers within their data centers, with a further 14% having plans to deploy them in the near future. Figure 1 illustrates an example of a typical blade enclosure. It has a total of sixteen blades in the front, eight on the top and eight on the bottom. The blades are cooled by up to ten fans in the back, five on the top and five on the bottom. The airflow generated by the fans is pulled through the blades towards the back of the enclosure with each fan contributing to the blade-level airflow rate. The enclosure and server architecture allows sharing of the cooling resources among the blades, and provide improved flexibility and configurability of IT resources. However, control of the fans in most cases are heuristic and zone based. Without an optimal fan control design, the over-provisioning of cooling capacity can be exacerbated in this architecture since a single hot spot can often drive the operation of a multiplicity of fans.

To address the thermal management of servers, this paper presents a model-based approach to managing fan power that is able to provide optimal cooling energy efficiency. We believe this is the first work to study such a model-based approach to fan control within servers. We make two main contributions in this paper.

First, we show how concepts from heat transfer theory can be combined with control system techniques to create accurate models for the cooling system in blade environment. Complex interaction exists within the blade environment among multiple variables including the power consumption and speeds of the fans, and the temperatures of the servers. Heat transfer theory allowed us to build models that can determine the individual and collective impact of adjusting fan speeds and varying workload on a server's temperature. The parameters were derived and the models were validated through experiments with only the sensors and knobs immediately available inside the blade servers that are very limited.

Second, we present an optimal fan speed controller based on the models, evaluated in a prototype system. Using the power and temperature models, we create a multiple-input multiple-output (MIMO) fan controller that can be built based on a power optimization problem that is tractable and allows for an online solution. The benefits of the models and controller were quantified through a real prototype that works with commercial, off-the-shelf hardware. We measure both power savings and impact on thermal performance with workloads gathered from real data centers. The results show that, without impacting temperature thresholds, our optimal controller can reduce cooling power by 20% compared to a zone-based integral feedback controller.

This paper is organized as follows. In the next section, we define the fan control as an optimization problem. The models needed to solve the optimization problem online are presented in Section 3. In Section 4, we discuss the implementation of the optimal controller that can solve the optimization problem in real time. Experimental evaluation and results are described in Section 5 before we conclude the paper in Section 6.

2 Problem Definition

Our goal of cooling management for blade servers is to minimize the total energy consumption by the fans of the enclosure while maintaining critical temperatures below their defined thresholds. Assume that the enclosure has **I** fans and **J** blades, then our goal is to

$$\min_{FS} \sum_i P_{F_i}, \tag{1}$$

where P_{F_i} is the power consumed by fan i ($i = 1, \ldots, I$), and FS is the vector of all the fan speeds. For proper thermal management, the temperature of each blade, T_j, should be maintained below T_{ref}, a reference threshold specified by the manufacturer:

$$T_j \leq T_{ref}, \quad \text{for any blade j.} \tag{2}$$

The cost function in Eqn. (1) and the constraints defined by (2) formulate a well-defined optimization problem. The time-varying and sometimes unpredictable nature of application demands require that the optimization problem be solved at runtime. Measurements for temperatures and actuators for fan speed tuning are available in most blade enclosures for real-time control. However, to solve the optimization problem, the controller needs models to determine the impact of actuator changes on the objective function and the constraints. For example, models are required to correlate fan speeds with both fan power and blade temperatures. Before we discuss how to design the controller that can solve the optimization problem in real time, we first present in the next section the models that can represent the complicated correlations between the actuators and sensors, and show how the model parameters can be identified through experiments using the sensors available in the servers.

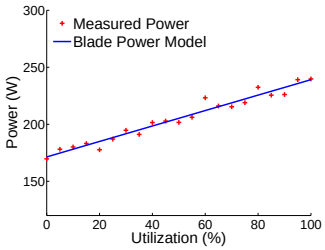


Figure 2. Blade Power Model

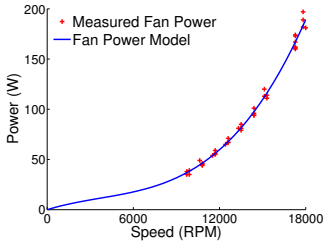


Figure 3. Fan Power Model

3 Models

Both power and temperature models are needed to solve the fan control problem. We first present in Section 3.1 the models that represent power consumption of servers and fans. The power of the servers needs to be modeled because it directly impacts server temperatures. Temperatures are significantly more challenging to model empirically than power. They are affected by multiple parameters including server workload, server inlet temperature, and component thermal properties. While a number of per-blade sensors, including memory and motherboard temperature, were available to determine a blade's internal temperature, we discovered, after performing sensitivity analysis, that T_{CPU_j} of the blade j, the processor temperature, was dominant. We therefore use T_{CPU_j} as a proxy for T_j, the temperature of the blade. The temperature models are described in Section 3.2 for both steady and transient states. Finally, we describe how to identify the model parameters through experiments and provide examples for validation of the models in Section 3.3.

3.1 Power Models

3.1.1 Blade power consumption

Although our goal is to minimize fan power, models for server power consumption are also required. Power consumed by the processors is a critical component of the thermal models that are to be derived, which is closely related with the power models of the blades. We therefore created a blade power model that characterizes the relationship between processor capacity utilization and blade power consumption. Our model is based on the fact that the power consumed by the processors is usually the dominant and most variable component of the server power [10].

Figure 2 shows the relationship between the processor utilization and the blade power consumption. A tunable CPU-intensive workload was used to gather the data from our experimental setup, described in Section 5.1. As shown by the trend line in Fig. 2, the power consumption of the blades can be approximated utilizing a linear model fit from the measured data as follows:

$$P_{B_j} = g_B * Util_j + P_{B,idle}, \quad \text{for any blade j,} \tag{3}$$

where $Util_j$ is the CPU utilization of blade j, g_B is a constant coefficient, and $P_{B,idle}$ is the power consumption of the blade when the CPUs are idle. This model is similar to those used in several other studies [11, 6, 12].

3.1.2 Fan power consumption

Fan power consumption is approximately a cubic function of the rotational speed of the rotor given in revolutions per minute (RPM) [13]. In our enclosure, the relationship was determined by manually setting the fan speed, FS, and then recording the power consumption of the individual fan, which is measured and reported by the system. Figure 3 shows the raw data obtained from this process. Note that there are frequencies, such as 17,100 RPM (95% of maximum speed) in Fig. 3, that the fans avoid to prevent resonance. In our experiments, the fan power is approximated using a 3rd-order polynomial, also shown as the solid line in Fig. 3:

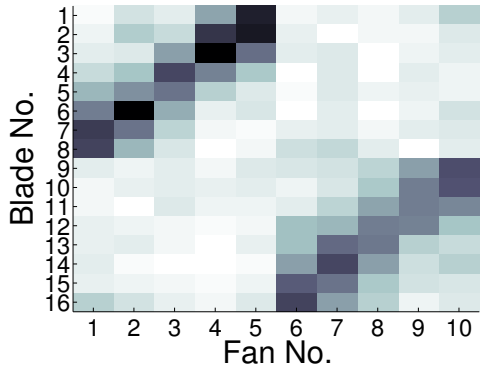
$$P_{F,i}(FS_i) = a_0 * FS_i^3 + a_1 * FS_i^2 + a_2 * FS_i \tag{4}$$

The parameters a_0, a_1 and a_2 were fit using the data samples and the origin. Note that the actual fan power is not exactly a cubic function, which can be due to the specific operation conditions of the fans, the air flows and the enclosure thermal architecture. But we believe the model derived from experiments is representative under the normal operation conditions of the enclosure.

3.2 Temperature Models

A number of challenges have to be addressed in order to create an effective model for the thermal environment of the enclosure shown in Fig. 1:

Zonal variations and complexity. There are many ways in which complexity in a shared system like the enclosure can arise. For instance, as mentioned earlier, fans are shared at the enclosure level and each fan contributes partially to the airflow across each blade. Figure 4, explained later in Section 3.2.1, quantifies this dependency between the fans and the blades via a heat map. The figure shows that while blade temperatures are most affected by fans closest to them, the degree of influence of each fan as well as the number of fans that can significantly affect a blade shows a lot of variation between blades. The details on how the values were derived will be described in Section 3.3.3. Apart from the air flow generated by the fans, an individual blade's temperature is also affected by the heat generated by the workloads it runs. Blade inlet air temperature (T_{amb}) can also differ between blades due to external physical effects such as the recirculation of hot air near the edges of an enclosure in a data center. A consequence of all these effects is that the model correlating temperature changes to fan speed changes is a complex I:J multivariable mapping function across four vectors (FS, T_{CPU}, $Util$,



The darker the shade of gray in the above heatmap, the stronger the effect each fan has on the corresponding blade. For example, changing the speed of fan 1 will have a much higher effect on the temperature of blade 8 than blade 1.

Figure 4. Blade:Fan Relationship

T_{amb}) rather than a simple function that can be reused across all blades and fans.

Inadequate coverage for thermal variables. While complex, as will be seen later, such an I:J multi-variable model can be formulated through the application of heat transfer theory. However, a key challenge is the absence of sensors in current systems that provide the level of detailed coverage for specific physical aspects of the system. An example would be sensors to detect blade-level volumetric airflow rate which affects heat sink thermal resistance. This requires additional work in the creation of models to ensure that the final relationship only involves variables that can be measured in current systems.

Steady-state versus transient behavior. When fan speeds or workload demands change, the change in power consumed is almost instantaneous. However, this does not hold true for temperature. Temporal delays exist between a change in fan speed or utilization and the corresponding effect on blade temperatures. While some fraction of the temperature change occurs within a few seconds, it can take a few minutes for temperatures to converge (assuming no other disturbances in fan speed or utilization occur). We derive in this section two types of models: a steady-state model, which defines the relationship between the actuators and sensors when both are at steady states, and a dynamic model, which represents the transient process.

3.2.1 Steady State Models

Our derivation of the steady state model for CPU temperatures leverages key concepts from heat transfer theory. Specifically, we use the concept of *thermal resistance* [14] to develop a model that relates CPU temperature to ambient temperature, heat generation, and volumetric air flow rate. Using two other relationships between volumetric airflow rate and fan speed, and between blade heat dissipation and CPU utilization, we obtain the required model relating the CPU temperature to the fan speed, the CPU utilization, and the ambient temperature (note that, as discussed earlier, all these terms are vectors).

We start by describing the thermal resistance, R, of a blade. In general, thermal resistance between two points is represented by the ratio of the temperature difference between the two points and the heat transferred per unit of time between those points. In our work, we are interested in the heat transfer between the CPU and the ambient air flowing through the blade. The thermal resistance is then defined as:

$$R_j = \frac{T_{CPU_j} - T_{amb_j}}{Q_j}, \quad \text{for any blade j,} \tag{5}$$

where Q_j is the heat transfered per unit of time between the CPU and the ambient air. Given that the processors are dominant in both the power consumption and the power variance of the blades, we use the CPU power consumption, P_{CPU}, as a proxy for Q. Similar to the blade power model (3), the CPU power is modeled as a linear function of its utilization as follows:

$$P_{CPU_j} = g_{CPU} * Util_j + P_{CPU,idle}, \quad \text{for any blade j.} \tag{6}$$

Note that the model may have different slope g_{CPU} from that in the server power model.

The temperatures and the heat transfered are external indicators of thermal resistance. Internally, thermal resistance depends on the material properties through which the heat is transfered, its geometry, fluid parameters like flow rate and turbulence, and interfacial effects between different materials (i.e. air flowing over a solid surface). In our work, we assume forced convection is the dominant mode of heat transfer from the CPU package (via the heat sink) and do not consider second order mechanisms, like radiation, that would add unnecessary complexity to the model. We also assume all of the heat generated by the CPU is transferred to the air through the heatsink. The thermal resistance for an individual blade is, therefore, approximated by

$$R_j = \frac{C_3}{\dot{V}_j^{n_R}} + C_4, \quad \text{for any blade j,} \tag{7}$$

where $\dot{V}_j$ is the volumetric air flow rate through blade j, C_3 and C_4 are constants related to the fluid and material properties of the air, the CPU package, and the heatsink. The parameter n_R defines the shape of the thermal resistance curve as a function of the air flow rate. It's primarily related to the level of turbulence in the flow which is a function of air velocity through the heatsink and heatsink design. All together, Equations (5), (6), and (7) represent the thermal model of a single blade processor.

Given that the air flow through one blade is an aggregate of the flows generated by all of the fans, and the rate of the air flow generated by each fan is approximately proportional to the fan speed, we can correlate the per-blade air flow to the fan speeds in the following manner:

$$\dot{V}_j = \sum_i \eta_{ij} \times FS_i, \quad \text{for any blade j,} \tag{8}$$

where η_{ij} is the correlation index between the speed of fan i and the air flow rate in blade j, and FS_i is the speed of fan i. The variation in the derived values for η_{ij} can be seen in Fig. 4. In light of Eqn. (5), (6), (7) and (8), the CPU temperature in steady state can be represented by a function of the processor utilization, the fan speed, and the ambient temperature as follows:

$$T_{CPU_j} = (g_{CPU} Util_j + P_{CPU,idle}) \left(\frac{C_3}{(\sum_i \eta_{ij} FS_i)^{n_R}} + C_4 \right) + T_{amb_j}, \quad \text{for any blade j.} \tag{9}$$

3.2.2 Transient Model

In transient heat transfer theory, the rate of change in the temperature of a device, like a CPU, is related to the rate at which heat is generated within the device and the rate at which heat can be transferred from the device to a cooling medium, like air in the present case. By conducting an energy balance among these three basic elements (rate of change of temperature, heat generation, and heat transfer), a differential equation can be formed that governs this dynamic relationship [14]. When heat transfer from a solid device, like a CPU package, is dominated by convection rather than by conduction within the package, the lumped capacitance method can be used to approximate the temperature of the device. Using this method, an energy balance over the device results in the following:

$$C_1 \frac{dT_{CPU,j}}{dt} = \frac{C_2}{R_j}(T_{amb,j} - T_{CPU,j}) + Q_j, \tag{10}$$

where C_1 and C_2 are constants related to the fluid properties of air, CPU package geometry, and material properties, and R_j is the thermal resistance given in Eqn. (7), t is the time, and Q_j is the heat transferred from the device. Model (10) shows how the transient temperature is affected by the workload, the environment and the fan speed (through R). At steady state, the CPU temperature can be derived from Eqn. (10), which is exactly the same as in (9).

The speeds of the fans will be tuned in discrete-time intervals through the fan controller, for which we need discrete-time models of the blade temperatures. Assume that the temperatures are sampled using a sampling interval of Δt. To get the discrete-time model, we assume that T_{amb} is constant in each sampling interval, or varies relatively slowly compared with $T_{CPU,j}$, which is usually true in practice. We also assume that the fan speeds are constant in each sampling interval, which means that the thermal resistance is fixed. Then the model defined in Eqn. (10) can be approximated as a first-order dynamic system in the form as following

$$\frac{C_1 R_j}{C_2} \frac{d\Delta T_j}{dt} + \Delta T = \frac{R_j}{C_2} Q_j, \tag{11}$$

where $\Delta T_j = T_{CPU,j} - T_{amb,j}$. A first-order system can be characterized by two parameters: a time constant τ and a steady-state gain G. The first parameter represents the time the temperature of the system takes to reach approximately 63% of its steady state upon a step change of Q. From the model we know that the time constant is $\tau_j = \frac{C_1 R_j}{C_2}$, which is a function of the thermal resistance, but independent of the heat transferred. The gain of the system is equal to $G_j = \frac{T_j(\infty)}{Q_j(\infty} = \frac{R_j}{C_2}$, which is again a function of the thermal resistance. When the temperatures are sampled in discrete time, the sampled-data system for (11) is as following [15]:

$$\Delta T_j(k+1) = e^{-\frac{\Delta t C_2}{C_1 R_j}} \Delta T_j(k) + (1 - e^{-\frac{\Delta t C_2}{C_1 R_j}}) \frac{R_j}{C_2} Q_j. \tag{12}$$

This model is the exact equivalent of the continuous-time model as in Eqn. (11), which means that they have the same time constants and the same steady-state gains. Under our assumptions on T_{amb} and FS, i.e, they are constant during the sampling intervals, Eqn. (12) represents the model (10) in discrete time. In our experiments, we have used (12) to predict the server temperatures as functions of the workload, the environment, and the fan speeds.

It is worthwhile to notice that, when Δt is small enough such that $e^{-\frac{\Delta t C_2}{C_1 R}} \approx 1 - \frac{\Delta t C_2}{C_1 R}$, the model in Eqn. (12) can be approximated as following:

$$\Delta T_j(k+1) = (1 - \frac{\Delta t C_2}{C_1 R_j}) \Delta T_j(k) - \frac{\Delta t}{C_1} Q_j, \tag{13}$$

which equals to the discretized model of (10) using Euler's method $\frac{dT}{dt} \approx \frac{T(k+1) - T(k)}{\Delta t}$. The model in Eqn. (13) is simpler than that in Eqn. (12). However, if the sampling interval Δt is not small enough, there will be significant error when Eqn. (13) is used to predict the temperature. That's why we have used Eqn. (12) instead of (13) in our experiments since the sampling interval in our case was in the same order of the time constant.

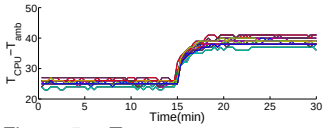


Figure 5. Temperature response upon step heat changes

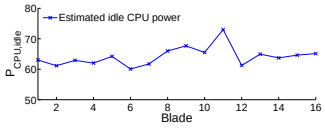


Figure 6. Estimated idle CPU power

The steady-state and transient models may not exactly represent the physical properties of the enclosure, which can be affected by many factors. However, first-order factors are expected to be captured by the models. The next section describes how the model parameters are identified through experiments run in a production environment together with a validation that shows these models can represent the state of the enclosure accurately enough for dynamic control.

In the models presented in this section, we assume that the constant parameters g_{CPU}, $P_{CPU,idle}$, n_R, C_1, C_2, C_3, C_4 are the same across the servers. This is reasonable for the servers in our test bed since all the blades are almost the same. However, this assumption is not necessary for the viability of our modeling approach.

3.3 Parameter identification and model validation

Given that the steady-state model (9) can be derived from the transient model (12), we focus on the models defined by Eqn. (5-7) and Eqn. (12). There are altogether 167 parameters in these models, including g_{CPU}, $P_{CPU,idle}$, n_R, C_1, C_2, C_3, C_4, and η_{ij}, for $i = 1,2,\ldots,10$ and $j = 1,2,\ldots,16$. To identify the values of these parameters, we ran a series of experiments on our test bed presented in Section 5.1. In this section, we describe what experiments we conducted, how the parameters were identified and how the models were validated.

3.3.1 Power consumed by the processor

As described previously, we assume that the heat transfered per unit time between the CPU and the ambient air is approximated by the power consumed by the CPU, which is a linear function of the CPU utilization, as defined in Eqn. (6). Note that the blade power is also represented as a linear function of the CPU utilization, as in Eqn. (3), the slope of which is derived from experimental data. The two slopes g_B and g_{CPU} are correlated. In the test bed, the blades have two sockets, each hosting two cores. In the experiments conducted for blade power model, all the cores were running workloads at the same CPU utilization level. Since Eqn. (6) is only modeling one CPU socket, we can set approximately

$$g_{CPU} = \frac{g_B}{2}. \tag{14}$$

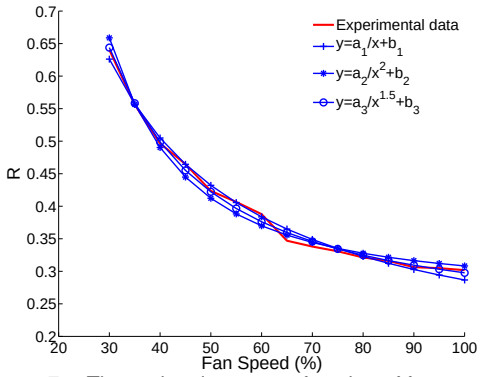


Figure 7. Thermal resistance as function of fan speed

The other parameter $P_{CPU,idle}$ can be estimated based on g_{CPU}. Note that the thermal resistance only depends on the air flow rate. For a given fan speed, the thermal resistance is independent of the amount of heat that is transfered. Then for any two heat dissipation levels Q_a and Q_b, corresponding to CPU utilization levels $Util_a$ and $Util_b$, we have $R_a = R_b$. From Eqn. (5), we can derive the $P_{CPU,idle}$ as following:

$$P_{CPU,idle} = g_{CPU}\frac{\Delta T_a Util_b - \Delta T_b Util_a}{\Delta T_a - \Delta T_b}, \tag{15}$$

where again $\Delta T = T_{CPU} - T_{amb}$.

We run one experiment to identify the parameter $P_{CPU,idle}$ based on the relation Eqn. (15), the metrics in the right hand side of which are known or measurable using the sensors readily available. At the beginning, the CPU was idle, i.e., with zero utilization, and the speeds of all the fans were set to 50%. After the server temperatures converged, the utilization levels were set to 100% by running a CPU intensive workload while the fan speeds were kept unchanged. Figure 5 shows the trajectories of ΔT for all the 16 servers. We then used the steady-state values of ΔT and the utilization levels to derive $P_{CPU,idle}$ based on Eqn. (15). Figure 6 shows $P_{CPU,idle}$ values for the 16 servers. All are similar with the exception of blade 11. $P_{CPU,idle}$ was set to the mean of those values in later experiments.

3.3.2 Thermal resistance

Since the temperature measurement is available, and the heat transferred can be estimated, thermal resistance can be identified using the relationship in Eqn. (5). The factor n_R is then derived from the relationship defined in Eqn. (5-8). Assume that all the fans have the same speeds (FS), then from Eqn. (7) and (8), we have

$$R_j = \frac{C_3}{(\sum_i \eta_{ij})^{n_R}}\frac{1}{FS^{n_R}} + C_4 \tag{16}$$

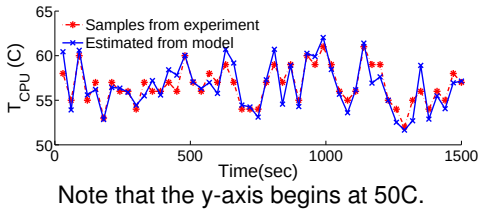


Figure 8. Model Validation

To estimate n_R, a series of experiments were designed. In each experiment, the speeds of all the fans were set to the same level for 40 minutes. The CPU utilization was set to zero for 20 minutes, and then pushed to 100% for another period of 20 minutes. One set of thermal resistance values for all the servers were then estimated from the steady-state temperature values. The experiments were repeated for different sets of fan speeds between 30% and 100%, with a step of 5%. Using all the thermal resistance samples and the fan speed values, we got the curves for the relation between R and FS. It was found that the curves, each for one blade, were close to each other and had approximately the same shape. In Fig. 7, the red line represents the mean of the R's as a function of the fan speed, from which we can approximate n_R through curve fitting.

As the fans are varied in speed during normal operation, it's possible that the flow characteristics could range from laminar, to transitional, to turbulent flow. An analysis of the thermal resistance curve shown in Fig. 7 indicates that the flow likely transitions from laminar to transitional flow at a speed of around 60%. To test this hypothesis, the Reynolds number of the flow through the blades was calculated using the hydraulic diameter of the blade gap in Fig. 1. (The gap between blades and the height of the blades was used in the hydraulic diameter calculation.) Transitional flow was estimated to occur in the blade gap at a fan speed of 56%, which is very similar to the location of the inflection in Fig. 7. It's desirable to use a single function to represent the thermal resistance curve over the entire operating range of the fans. We tried different numbers for n_R, and found that the curve with $n_R = 1.5$ matched the best with the experimental data especially in the lower and higher ends. This is consistent with flow that undergoes transition within the range of fan speed operation. (Note that n_R should range between 1 and 2 with higher values associated with more laminar flows.) This value was utilized in later experiments to determining the remaining parameters.

3.3.3 Parameters for transient models

With g_{CPU}, $P_{CPU,idle}$, and n_R identified, the remaining parameters were estimated through a system identification experiment, in which the speeds of all the fans and the CPU utilization levels on all the blades were varied randomly every 30 seconds. The experiment lasted for a few hours. The metrics including the CPU utilization, the ambient temperatures, the CPU temperatures, and the fan speeds were collected in real time. The data samples and the temperature model defined by Eqn. (7), (8) and (12) were then fed into the *nlinfit* tool in Matlab, which fits through nonlinear regression the values of all the remaining parameters except C_3. Note that C_3 and the η's cannot be identified independently. In our experiment, C_3 was set to 1.0 which is arbitrary.

A few tests for model validation implied that the models and the parameters identified are reasonable. Figure 4 provides one validation of the models. In this case, and as described in Section 3, nearby fans having a more significant impact on blade temperatures than those located farther away matches our intuition. In Fig. 8, a separate set of data samples from experiments were used for model validation. The model-predicted temperatures accurately captured the impacts of the workloads on the measured temperatures, with the prediction errors less than 5% of the measured ones. Note that only data in discrete time points were sampled or predicted, and the lines are only to show the trends.

4 Controllers

The models developed in the previous section do imply challenges in the fan controller design due to zonal variations and complex interdependencies between the fans and the blades, and the presence of nonlinear relationships, for instance, between the fan speeds and the temperatures. However, the models also provides us intuitions to decompose the complex optimization problem into multiple sub-problems that are easier to be solved. We present in this section one such optimal fan controller, called FC, that we implemented in our prototype and evaluated through experiments.

4.1 Optimal fan controller

The objective of the fan controller (FC) is to minimize the total fan power while satisfying the cooling requirements of all the blades by periodically adjusting the fan speeds. Since the fans are shared among the blades, the controller needs to consider all the fans and the blades simultaneously. One immediate solution for the fan controller is to solve the following optimization problem at the end of each control interval k:

$$\min \sum_i P_{F_i} \tag{17}$$

$$T_{CPU,j}(k+1) \leq T_{ref}, \quad \text{for each blade j} \tag{18}$$

$$LB_i \leq FS_i \leq UB_i, \quad \text{for each fan i} \tag{19}$$

The objective function aims to minimize the sum of the power consumptions (P_{F_i}) of all the fans. The first constraint ensures that the CPU temperature for each blade j in the next interval $k+1$ remains below T_{ref}. The second constraint ensures that

each of the fan speeds does not exceed its lower bound (LB) or upper bound (UB). The optimal solution will be the fan speeds to be configured for the next control interval. However, there are a few challenges for this optimization problem:

1. The cost function and the first set of constraints are nonlinear, which preclude the use of efficient optimization techniques such as linear programming.
2. The transient thermal model in Eqn. (12) is needed to predict $T_{CPU_j}(k+1)$, based on the current CPU temperature $T_{CPU_j}(k)$, the ambient temperature $T_{amb}(k)$, and the estimated CPU power consumption P_{CPU_j}. To represent the constraints in terms of fan speeds, we need to express the thermal resistance (R_j) as a function of all the fan speeds FS (see Eqn. (7) and (8)). These nonlinear constraints preclude the use of most convex optimization tools.
3. This problem may not be feasible since the fan speeds are physically upper bounded and such that the lowest temperatures that are achievable could be higher than the thresholds. The feasibility issue has to be carefully considered in each control step.

Some facts observed on the models can help us to deal with the challenges. Simple analysis shows that the first-order derivative of $\Delta T_j(k+1)$ defined in Eqn. (12), w.r.t. R_j, is positive. That is, the temperature $T_{CPU,j}(k+1)$ is a monotonically increasing function of R_j. Assuming that the $T_{amb}(k+1)$ is the same as in previous interval[1], the constraint (18) can then be converted as a bound on R_j. Using the model (7), the constraint can further be converted to that on $\dot{V}_j$, which actually represents the cooling demand of blade j in terms of air flow rate.

Based on above facts, we define the optimal fan controller (FC) as following that solve the optimization problem in two steps:

Step 1: For each blade j, solve the local optimization problem

$$\min \dot{V}_j \tag{20}$$

$$\text{s.t. } T_{CPU,j}(k+1) \leq T_{ref} \tag{21}$$

This problem is to find the minimum cooling demand of the blade that can meet the temperature requirement. For discussion followed, we assume it is $\dot{V}_j^o$.
Note that the fan speeds are upper bounded (by UB), which means that the air flows available to the blades are also bounded, and such that $\dot{V}_j^o$ may be infeasible. Define $\dot{V}_{jM} = \sum_i(\eta_{ij} UB_i)$, the maximum air flows available for the blade j, then the minimum but achievable flow rate in blade j should be as following:

$$\dot{V}_j^* = \min(\dot{V}_{jM}, \dot{V}_j^o). \tag{22}$$

Step 2: With the minimum air flow demand $\dot{V}_j^*$ from each blade j, solve the global optimization problem

$$\min_{FS} \textstyle\sum_i P_{F_i}(FS) \tag{23}$$

$$\textstyle\sum_i \eta_{ij} FS_i \geq \dot{V}_j^*, \quad \text{for each blade j} \tag{24}$$

$$LB_i \leq FS_i \leq UB_i, \qquad \text{for each fan i} \tag{25}$$

The problems in the two steps are more tractable than that defined by Eqn. (17-19). In the first step, given that the temperature $T_{CPU,j}(k+1)$ is a monotonically decreasing function of the flow rate $\dot{V}_j$, the problem can be solved through efficient search algorithms, for instance, binary search, started from the range defined by $\dot{V}_{jm}$ and $\dot{V}_{jM}$, where $\dot{V}_{jm} = \sum_i(\eta_{ij} LB_i)$. The problem in the second step is a convex optimization problem [16] with a polynomial but convex objective function and linear constraints. It can be solved using standard tools. In our prototype, a Python-based software package, `cvxopt` [17], was used.

4.2 Integral Fan Controller

This section contains a brief description of an alternate fan controller that will be used to compare the FC against. Unlike the FC, which is a predictive controller, the Integral Fan Controller (IFC) is a simple reactive controller that increases or decreases the fan speeds based on the error between the temperature reference and current measurement. It is similar to commercial controllers used in industry today. The IFC measures the maximum temperature of the blades and compares it to the reference threshold T_{ref}. It then either increases or decreases the fan speed depending on whether the measured temperature is higher or lower than that reference. Given that each blade is mainly affected by the fans present in its row, as shown in Fig. 4, the IFC separately controls the fan speeds on a per-row level based on the maximum blade temperature recorded for that row.

5 Evaluation

5.1 Experimental Setup

To evaluate the performance of the fan controllers, we used an HP c7000 BladeSystem enclosure with 16 ProLiant BL465c server blades and 10 fans. As shown in Fig. 1, the blades and fans within this enclosure are equally divided into two rows. Each blade is equipped with two AMD 2216 HE processors with two cores each, and comes with seven pre-installed temperature sensors. Three sensors are located in the CPU region, two for the memory regions, one near the front to measure the inlet air temperature, and one that measures the motherboard temperature.

[1] In our experiments, the difference of the ambient temperatures in two consecutive intervals were at most 1C.

The enclosure also contains an Onboard Administrator (OA), an embedded module running Linux, that provides integrated enclosure management. The OA allows us to record all the temperature readings as well as the power used by the entire enclosure and each individual fan. It also allows us to control the speed of individual fans between 3,000 and 18,000 RPM.

The fan controllers were run on a separate workstation, connected to the OA for temperature information and for fan speed control. While the enclosure can handle higher temperatures, we set $T_{ref} = 65$C and the minimum fan speed (LB) to 4,000 RPM to ensure equipment safety while conducting our experiments. The sampling and control interval Δt was set to 30 seconds due to the actuation delay for fan speed configuration through the software and firmware stacks.

For comparison purpose, we run two experiments, in which the fan speeds were under control of the optimal fan controller (FC) and the feedback controller (IFC) respectively. To have fair comparison, we tuned the gain parameter of IFC so that the thermal performance, or the temperature violation levels between the two controllers are close to each other.

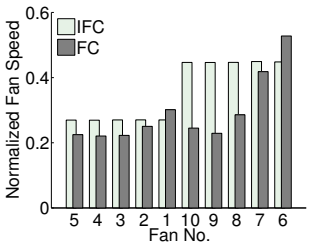


Figure 9. IFC vs. FC: Fan Speeds

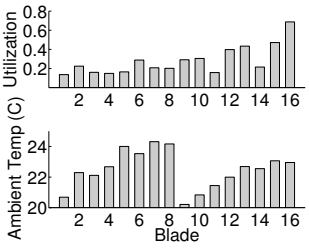


Figure 10. Utilization and Temperatures for both IFC and FC

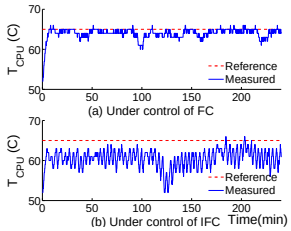


Figure 11. IFC vs. FC: Temperature trajectories of Blade #1

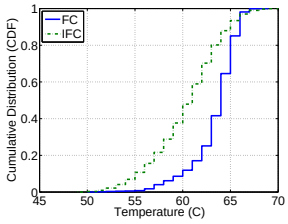


Figure 12. IFC vs. FC: Temperature distributions of all blades

5.2 Benchmarks

To obtain a realistic estimate of the possible savings of our system, we used traces gathered from 64 servers in real data centers running e-commerce and database workloads and are representative of a traditional IT environment found in large corporations. Among the 64 traces, 80% of them have an average utilization lower than 24%. While this low utilization is typical in data center environments, it does not mean that the resource usage is uniform over the entire time period. Our analysis of the traces showed that not only were they bursty, but they also exhibited periodicity.

In our experiments, each blade hosted four Xen [18] virtual machines, and inside each virtual machine, a workload generator tool was utilized to replay one of the 64 traces. While the traces were gathered over a period of a number of days, in the interests of time, our experiments used a representative four-hour-long segment from the busy periods.

5.3 Results: Fan Power

On average, the total power consumed by the 10 fans was 213 watt when they were under control of the feedback controller (IFC). It was 172 watt while under control of the optimal controller (FC). Compared with IFC, the FC controller reduced fan power usage by about 20%.

We examine this result in detail in Fig. 9, which shows the time average observed fan speeds for the IFC and FC. Overall, more power was consumed with the IFC as the fans under its control, with the exceptions of 1 and 6, were driven to higher speeds than those of the FC. Unlike the IFC, where the fans in the same row run at the same speed, the FC varies the fan speeds with a much finer granularity and is able to provide "on-demand" cooling to the blades. Figure 10, displaying the average utilization and ambient temperature of each blade, provides further insight into these results[2]. For example, blades 15 and 16 had the highest utilization and were located in a region of the enclosure with higher ambient temperatures. Given that fans 6 and 7 have the strongest cooling effect on the two blades, as shown in Fig. 4, their speeds, shown in Fig. 9, were higher than the rest.

While the savings in cooling power observed is significant, we consider the results to be conservative due to the fact that the enclosure used for our experiments is over-provisioned in cooling resources and uses very low-powered CPUs. Average fan power consumption, therefore, tended towards the lower half of the fan power curve shown in Fig. 3. Given the nonlinearity in fan power consumption, we expect that for future generations of blade servers that more fully utilize the available cooling resources, the optimal fan controller will provide even greater savings.

5.4 Results: Temperature Control

Besides of power consumption, thermal management performance should be considered when evaluating the control methods. Figure 11 shows the temperature trajectories of the blade #1, when the blades were under control of FC and IFC respectively. (Note that the temperature sensors returned only integers

[2]Note that the results shown in Fig. 10 are identical for both the FC and IFC scenarios as the workload is fixed and there were no changes in the data center environment during these experiments.

which were in degree C.) With the optimal fan controller FC, the temperature was maintained close but below the reference, 65C, most of the time. However, with the IFC feedback controller, the temperature was kept further below the reference, and oscillating in a large range. Similar difference can be found from the temperature trajectories of all the other blades. The difference between the temperature trajectories is due to that of the two controllers: individual cooling demand is considered by FC all the time, while only the highest temperature of the blades in the same row is under control of IFC. Figure 12 provides statistics on the temperature samples of all the blades. It is found that 98% of the samples are not higher than 66C when using FC, and the number is 97% when using IFC. These numbers mean that the two controllers are comparable in term of maintaining the temperatures below the thresholds. However, using FC, 57% of the samples are in between 64C and 66C, while it is only 17% when using IFC. With finer control granularity and more knowledge on the blades, the FC minimized the energy consumption by the fans by pushing each of the blade temperatures to their limits as much as possible.

6 Conclusion

In summary, this paper has introduced an optimal and predictive MIMO fan controller for thermal management of servers. To the best of our knowledge, this is the first model-based fan controller for blade server environment. Using fundamental concepts from heat transfer theory, we developed powerful models that can capture complicated correlation among temperatures, workloads, and fan speeds through simple experiments in a real system and by using measurements provided by the system itself. We proposed a hierarchical fan controller that can simplify the optimization problem and maximize the energy efficiency of the fans by meeting the cooling demand of the individual servers. Experimental results show that our controller can in real time save significant amount of cooling power while maintain thermal safety for the servers.

We believe that, both our modeling and control approaches can be extended to tackle with other temperature metrics, e.g., those of memory, and other systems in larger scale, e.g., racks and even data centers, which is a part of our on-going work. We are also working on integration of the predictive approach and feedback control for fan control as we believe that it can provide even better control performance.

REFERENCES

[1] U.S. Environmental Protection Agency (EPA), 2007. Report to congress on server and data center energy efficiency, public law 109-431, Aug.

[2] Greenberg, S., Mills, E., Tschudi, B., Rumsey, P., and Myatt, B., 2006. "Best practices for data centers: Results from benchmarking 22 data centers". In Proceedings of the 2006 ACEEE Summer Study on Energy Efficiency in Buildings.

[3] Patel, C. D., Bash, C. E., Sharma, R., Beitelmam, M., and Friedrich, R. J., 2003. "Smart cooling of data centers". In Proceedings of IPACK'03, The Pacific Rim/ASME International Electronic Packaging Technical Conference and Exhibition.

[4] Bash, C., and Forman, G., 2007. "Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center". In Proceedings of the USENIX Annual Technical Conference, pp. 363–368.

[5] Bash, C. E., Patel, C. D., and Sharma, R. K., 2006. "Dynamic thermal management of air cooled data centers". In Proceedings of the 10th International Conference on Thermal and Thermomechanical Phenomena in Electronics Systems (ITHERM), pp. 445–452.

[6] Heath, T., Centeno, A. P., George, P., Ramos, L., Jaluria, Y., and Bianchini, R., 2006. "Mercury and freon: Temperature emulation and management for server systems". In Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 106–116.

[7] Moore, J., Chase, J., Ranganathan, P., and Sharma, R., 2005. "Making scheduling "cool": Temperature-aware workload placement in data centers". In Proceedings of the USENIX Annual Technical Conference, pp. 61–75.

[8] IDC, 2008. Worldwide quarterly server tracker, Q4 2007, Feb.

[9] Data Center Users' Group, 2008. Spring 2008 Data center users' group survey results. `http://datacenterug.org/`.

[10] Heath, T., Diniz, B., Carrera, E. V., Jr., W. M., and Bianchini, R., 2005. "Energy conservation in heterogeneous server clusters". In Proceedings of the ACM Symposium on Principles and Practice of Parallel Programming (PPOPP), pp. 186–195.

[11] Fan, X., Weber, W.-D., and Barroso, L. A., 2007. "Power provisioning for a warehouse-sized computer". In Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA '07), pp. 13–23.

[12] Raghavendra, R., Ranganathan, P., Talwar, V., Wang, Z., and Zhu, X., 2008. "No "power" struggles: Coordinated multi-level power management for the data center". In Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 48–59.

[13] Jorgensen, R., ed., 1983. *Fan Engineering*, 8th ed. Buffalo Froge Company.

[14] Özisik, M. N., 1985. *Heat Transfer: A Basic Approach*. McGraw-Hill Companies.

[15] Franklin, G. F., Powell, J. D., and Workman, M., 1998. *Digital Control of Dynamic Systems*, 3 ed. Addison-Wesely.

[16] Boyd, S., and Vandenberghe, L., 2004. *Convex Optimization*. Cambridge University Press.
[17] cvxopt. `http://abel.ee.ucla.edu/cvxopt/`.
[18] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A., 2003. "Xen and the art of virtualization". In Proceedings of the 19th ACM Symposium on Operating Systems Principles, pp. 164–177.