A CDF-Based Tool for Studying Temperature in Rack-Mounted Servers

Jeonghwan Choi, Youngjae Kim, Anand Sivasubramaniam, *Member*, *IEEE*, Jelena Srebric, Qian Wang, *Member*, *IEEE*, and Joonwon Lee

Abstract—Temperature-aware computing is becoming more important in the design of computer systems as power densities are increasing and the implications of high operating temperatures result in higher failure rates of components and increased demand for cooling capability. Computer architects and system software designers need to understand the thermal consequences of their proposals and develop techniques for lowering operating temperatures to reduce both transient and permanent component failures. Until recently, tools for understanding the temperature ramifications of the designs of the server and the rack have been mainly restricted to the industry for studying packaging and cooling mechanisms and they have been mainly concerned with the static thermal characteristics of computer systems. Recognizing the need for such tools, there has been recent work on modeling temperatures of processors at the microarchitectural level, which can be easily understood and employed by computer architects for processor designs. However, there is a dearth of such tools in the academic/research community for undertaking architectural/systems studies beyond a processor—a server box, rack, or even a machine room. In this paper, we present a detailed three-dimensional Computational Fluid Dynamics-based thermal modeling tool, called *ThermoStat*, for rack-mounted server systems. We conduct several experiments with this tool to show how different load conditions affect the thermal profile and to also illustrate how this tool can help design dynamic thermal management techniques. We propose reactive and proactive thermal management for rack-mounted server and isothermal workload distribution for rack.

Index Terms—Simulation, energy-aware systems, power management, thermal modeling.

1 Introduction

Growing power densities are making thermal consideration a first-class citizen in the design and deployment of next-generation servers and data centers. We are already witnessing the limitations imposed by power consumption within individual chips, where the generated heat is forcing processor vendors to scale back frequency growth rates and to resort to alternative techniques for pushing the performance envelope. Similar challenges are also being encountered in the disk drive market, where thermal issues are restraining sustained growth in data rates [1], [2]. Technological advances in microprocessor design have resulted in high device density and performance. In current generation chips, power density and consequent temperature (*"Hot Spot"*) are becoming severe problems mainly due to nonideal scaling. Higher peak and average temperatures lead to lower lifetimes at the chip and system level. It has been reported that increasing operating temperature by $10\text{-}15^{\circ}\mathrm{C}$ [3] in electrical circuits leads to a halved lifetime. In addition, the cooling and packaging cost for heat dissipation increases with the total power, as well as the peak on-chip temperatures [4], [5]; in fact, the cost increase gradient is steeper at higher values of power and power density. As we step out of these individual components, thermal issues are starting to mandate sophisticated techniques for cooling dense server blades and rack-mounted systems, which are becoming more prevalent in machine rooms and data centers. Across this spectrum of granularity, high temperatures can lead to unreliable operation of components and even accentuate their failure rates. Deploying sophisticated cooling systems for machine rooms to accommodate the growing power densities can require a substantial initial investment, in addition to the environmental concerns and high cost of running/powering high-capacity Computer Room Air Conditioning (CRAC) systems.

All of these factors point to the need for designing systems for the average/common case behavior, with dynamic thermal management (DTM) techniques (DTM) stepping in when thermal emergencies are encountered. Such a design philosophy requires an in-depth understanding of several interrelated cross-domain topics covering computer architecture/circuits, systems software, thermodynamics, fluid dynamics, packaging, etc. Furthermore, it requires cross-cutting tools, where one can study different interactions, e.g., workloads, temperature, air flow, and system/room geometries. Until recently, the two domains—architecture and packaging—have been operating more or less independently when designing systems, with each working under a given set of constraints

- *J. Choi, Y. Kim, and A. Sivasubramaniam are with the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802. E-mail: {jechoi, youkim, anand}@cse.psu.edu.*
- *J. Srebric is with the Department of Architectural Engineering, Pennsylvania State University, University Park, PA 16802. E-mail: jsrebric@engr.psu.edu.*
- *Q. Wang is with the Department of Mechanical Engineering, Pennsylvania State University, University Park, PA 16802. E-mail: quw@psu.edu.*
- *J. Lee is with the Department of Computer Science, Korea Advanced Institute of Science and Technology, Kusong-Dong, Yusong-Gu, Daejon 305-701, South Korea. E-mail: joon@kaist.ac.kr.*

Manuscript received 3 Jan. 2007; revised 25 July 2007; accepted 10 Dec. 2007; published online 20 Mar. 2008.
Recommended for acceptance by J. Antonio.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0002-0107.
Digital Object Identifier no. 10.1109/TC.2008.52.

0018-9340/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

from the other. Instead, designing such adaptive systems and DTM techniques requires a closer harmony between these domains, with tools that each can use to study their interactions with the issues from the other domain. We are witnessing growing evidence of this trend, with recent thermal modeling tools at the individual component level (for example, [6], [7] for processors and [1] for disk drives), which are being used by system designers for architectural/software innovations [6], [8], [9], [10], [11], [7] to address thermal issues. However, there are few such tools available for a complete system—either a single server or a full rack. Thermal modeling tools for servers and racks are extensively used in the industry, mainly for packaging studies and rating machine ambient temperatures, most of which are proprietary and are not readily available to the academic/research community.

A recent utility [12] has been proposed to emulate the temperature of certain specific points of a server using simple flow equations. Our approach, on the other hand, uses Computational Fluid Dynamics (CFD) simulation to provide a complete 3D profile of the temperature within the system. We present a server and rack-level thermal modeling tool called *Thermal Statistics (ThermoStat)*, which can be customized for a given deployment with different geometries, placement of components (1U slots, processors, disks, network cards, etc.), their power consumption, cooling mechanisms (placement and CFM of fans, etc.), and inlet air conditions. Together with providing steady state temperatures, the tool can also provide details on how the temperatures change in the 3D space when specific system events (e.g., power dissipation of a processor changes due to change in dynamic activity or voltage/frequency, a fan breaks down, the external air temperature suddenly increases because of a door being open, or CRAC breakdown, etc.) occur and how long such a change takes. It can thus be integrated with other performance-power simulators [13], [14], [15] used by the architecture/systems community for integrated studies or can be run in standalone mode after obtaining the required values from those simulators. Such integration would involve an iterative procedure for the exchange of boundary conditions, where the power output from the system-level simulators would be used in ThermoStat simulations, while the environmental temperature and convective heat transfer parameters from ThermoStat would be used in the system-level simulators. We have modeled 42U rack-mounted servers using this tool and we have validated it by comparing the predictions with temperature readings from 29 sensors deployed both within different servers of this rack, as well as different points in the rack itself, and that from an infrared thermal camera.

Just as packaging engineers use such tools for figuring out how the underlying components can best be put together, ThermoStat can be used in static settings to determine 1) where components (processors, memory, NICs, disks, etc.) need to be located within a server, where fans (and their CFMs) need to be placed, and 2) how one can place the servers, network switches, and disk arrays within a rack and design the airflow for a rack. In addition, it can also be used to study how systems/components need to scale in the future (as in [1]) and to understand how the ramifications of any proposed enhancements on the power density impact system design. More importantly, we anticipate the use of a tool such as ThermoStat for designing and evaluating different "what-if" DTM techniques as described in the following:

- Until now, DTM has been restricted to one component at a time, e.g., a processor makes its decisions (say, DVS), independent of other components. However, with denser packaging, components are becoming more interrelated, i.e., the power dissipated by the processor can impact the temperature of the NIC, disk, graphics card, etc. Consequently, a more global strategy for thermal management may be necessary, which has not been considered until now because of the lack of sufficient tools. Information on fluid flow which is essential for undertaking such studies is typically unavailable on an infrastructure that only provides temperature sensors (which is the case on an actual platform).
- Proactive thermal management can be a better alternative than a purely reactive option in several situations. For instance, rather than waiting for the temperature to reach a threshold before taking remedial actions after a temperature impacting event (e.g., fan breakdown), better runtime mechanisms could be employed if we knew 1) whether the temperature will, in fact, reach emergency proportions and 2) how long it would take us to reach that point. Proactively, one could employ different options such as migrating computations and employing DVS for lower stall times and/or lower durations in emergency operating conditions. The tool can help identify which events can lead to emergencies, how long it would take us to get there, and what the best recourse for those conditions is.
- Workload distribution to a rack-mounted system with reference to peak temperature affects the thermal profile of the rack and data center management. Uneven temperature distribution around the rack prevents the data center management system from accommodating more workload. Isothermal workload distribution can be one possible solution for this problem. ThermoStat is a useful tool for enabling such workload placement policies. We can build a thermal profile database for various workload distributions and refer to this database to get isothermal workload distributions for a given workload requirement or temperature threshold.
- Such a tool can also be a useful building block in a larger infrastructure/setting to determine whether the rewards of the service provided at a certain level justify the cost of operating/cooling these systems and to modulate the level of service accordingly. With growing energy costs, revenue-based thermal management becomes extremely important for next-generation data centers [16], [17].

The rest of this paper is organized as follows: An underlying philosophy behind ThermoStat's design, details of CFD modeling, and validation results are discussed in Sections 3 and 4. In Section 5, we present the use of ThermoStat for studying thermal behavior of rack-mounted server. Illustration of the usage of ThermoStat for DTM at a rack-mounted system and a system workload placement

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

policy are given in Section 6. Finally, Section 7 concludes with directions for future work.

2 Related Works

As explained earlier, there have been recent developments in the availability of thermal modeling tools for architectural studies in the academic/research community. One such tool is HotSpot [6] for microprocessors, which models temperature using thermal resistances and capacitances derived from the layout of microarchitectural structures that has been validated using finite-element simulation. Rather than detailed thermal simulators for processors, quick estimation using convective energy dissipation techniques are used after calculating a processor's energy consumption using event counters in [18], [7]. Such estimation has been used for developing temperature aware scheduling [7]. There have also been thermal modeling studies for individual disks [1] and disk arrays [19], with the former providing a tool that also integrates with a disk performance simulator for architectural studies. These tools, which allow integrated performance and power/thermal studies, have been facilitating research contributions [20], [6], [21], [22], [23], [11], [24], [25] in the architecture community to reduce power/temperature.

All of these tools are useful when studying and optimizing individual components. In addition to specific components, in this paper, we are also interested in studying complete server systems, where there could be interactions between different components. For instance, it is not clear how well these models are suited when we need to find out how long it takes (if at all) for the temperature to reach emergency levels after a fan breaks down. Furthermore, it would also be useful to have a unified framework/tool for studying both static issues (e.g., where different components should be placed and how much cooling capacity is needed) and DTM techniques. A recent tool [12] proposes using simple equations to calculate temperatures at very specific points in the server system. While this approach suffices for certain simple "what-if" questions, as suggested in [12], a CFD-based model is needed for a more holistic examination of the system under a wider spectrum of static (e.g., where components like fans should be placed) and dynamic (e.g., how long would it take the temperature to reach a threshold upon fan failure and what thermal management technique provides the best recourse upon emergency). We further elaborate on these issues in this paper. Fluid flows need to be modeled accurately to determine where components need to be placed and to understand complete system interactions. The importance of cooling high density data centers/machine rooms has attracted considerable interest recently [26], [27], [28], [29]. Most of these studies (e.g., [30], [31], [32], [33], [34], [35], [36]) have looked at this problem from an engineering perspective of designing CRAC and other cooling systems, placement of racks in machine rooms, etc., most of which use CFD models. For instance, Patel et al. [32] point out that heat recirculation is a limiting factor in existing cooling systems and propose using heat exchangers in the ceiling. The impact of CRAC failures on static provisioning has also been studied using CFD models [35]. From the computer science/systems perspective, researchers are starting to use CFD models for workload placement [37], [38] across racks of a machine room and are balancing the temperature across these racks [39].

We intended to provide the tools for bridging the gap between these two granularity of thermal models—those at the individual component level (within processors or disks) and those at the machine room level (comprised of multiple racks)—for conducting both static and DTM studies. Though such tools exist in industry for studying packaging/cooling systems, we intend to provide a customizable and easily usable infrastructure for the academic/research community to allow integrated performance-power-temperature studies for further architectural/systems innovations.

3 ThermoStat: A Tool for System-Wide Thermal Study

In this section, we introduce ThermoStat, a tool for studying the thermal profile of rack-mounted servers. We introduce CFD and specify the model and method used in ThermoStat. Real temperature measurement results by a thermal sensor are described to validate ThermoStat.

3.1 Rationale for Methodology

3.1.1 Simulation versus Live System

There are several motivating reasons driving the need for a thermal profile simulator as compared to just living with temperature sensors on an actual platform:

- Sensor measurements can be inaccurate. In fact, the thermodynamics community usually places more emphasis on CFD simulations than just sensor measurements due to their low resolution and poor precision since these sensors may not necessarily measure the temperature at a single point in space. Furthermore, transitional effects can cause short-term fluctuations and the sampling needs to be done at extremely fine resolution to get confidence in the measured values.
- In addition to temporal variations, there can be high spatial variances in temperatures as well. In fact, we have noticed that temperatures can change as much as 16°C when we move even just a few centimeters in certain spatial regions of our system. Consequently, sensor placement becomes a very critical issue. We wish to point out that sensors need to be placed not only at the points where thermal emergencies need to be monitored but also at other spatial regions that can affect the temperature at these points (which may be needed for proactive control). Densely filling the three-dimensional space with temperature sensors is an infeasible unattractive option.
- Creating emergencies to study thermal profiles and associated optimizations on an actual system can be a very costly process: Components can break down. These experiments may also need to be conducted multiple times (with hopefully repeatable results) for statistical confidence. Furthermore, one may need to perform these thermal studies at the design stage, before the physical realization.

Some of these issues—such as the last point about the cost of building and conducting extensive tests on actual

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

platforms—are not unique to thermal modeling and simulators have traditionally been used to address such concerns. Even though field testing of the ideas on an actual platform is eventually needed to verify their benefits on full-fledged workloads, simulators are still very useful vehicles for developing, refining, and comparing innovative proposals. Consequently, simulators have been the potter's wheels of computer architects and have evolved over the years to different degrees of sophistication to answer "what-if" questions at various stages of design. We use a similar philosophy in opting for a simulation-based methodology for ThermoStat.

3.1.2 Needs for CFD Simulation

There could be a different granularity at which one could simulate the system under consideration, each with associated performance-accuracy trade-offs. For instance, in the widely used SimpleScalar simulator, there are several simulation options, two of which are a purely functional simulator (sim-fast) or a more detailed microarchitectural simulator (sim-outorder). We could even have finer resolution models going down to the RTL, gate, or even layout levels. As we go to a finer resolution, the accuracy of the model improves, though the cost (time) of simulation increases. We believe that understanding the complex fluid flows within the servers of a rack requires detailed modeling of its geometry and the position/parameters of power sources and fans. Such a level of modeling is usually done through CFD simulations.

We wish to point out that different simulation/modeling techniques have different pros and cons and their merits really depend on the intended use of tools developed using these techniques. For instance, Bellosa et al. [18], [7] use a simple set of differential equations to model the convective heat flow out of a processor based on Newton's law of cooling and obtain the processor temperature. This technique is simple and easy to compute, with the advantages of being able to model the temperatures in real time. It is also a fairly good model when the intention (as in the case in this work) is to simply understand and modulate the processor temperature as a function of its load. However, such simple models may not suffice when studying complete systems with other external events affecting the temperatures. For instance, one may be interested in finding out how long a window exists before the temperature reaches emergency levels once a fan breaks down. One may need detailed fluid flow models, typically influenced by several fans and several gradients of temperature differences on today's servers, to understand such complicated interactions.

One drawback of a detailed CFD model, which is analogous to going finer than a functional-level architectural simulator, is the time involved for such detailed simulations (which is discussed further in Section 7). We still believe in using a CFD-based approach for ThermoStat for the following reasons. First, just as in architectural simulators, we can run these CFD simulations in an *offline* manner to answer different "what-if" questions to understand the spatial and temporal temperature interactions between different components (a characterization study for a target platform). Such information can be used to compare between different server design/layout choices and/or even suggest better designs. Second, these simulations can again be run in an offline fashion to find out the suitability and reaction times of different DTM techniques. It is conceivable that a number of common/important thermal emergencies can be captured by these offline simulations and the (parameterized) remedial actions to be taken can then be stored in a database for consultation at runtime. Finally, ThermoStat can be a way of validating other temperature measurement (using sensors) or modeling (as in [18], [7]) techniques and can be used in conjunction with those to develop hybrid multiresolution models.

3.2 CFD Modeling

3.2.1 Governing Equation

For a spatial domain (a rack and/or a server system), CFD solves the governing transport equations represented in the following conservation law form:

$$\frac{\partial \rho \phi}{\partial t} + \frac{\partial \rho U_j \phi}{\phi \partial x_j} = \frac{\partial}{\partial x_j}\left(\Gamma_{\phi,eff}\frac{\partial \phi}{\partial x_j}\right) + S_\phi, \tag{1}$$

where the general variable ϕ stands for different parameters such as mass, velocity, temperature, or turbulence properties, ρ is the fluid (air) density, t is the time for transient simulations, x_j is a coordinate x, y, or z when j is $1, 2$, or 3, U_j is the velocity in the x, y, or z direction, Γ is the diffusion coefficient, and S is the source for a particular variable such as the heat flux emitted from the rack components when ϕ is the air temperature. The four equation terms represent the transient, convection, diffusion, and source parts of transport phenomenon taking place in the spatial domain/extent.

The transport equations represent a system of partial differential equations that are coupled together and need to be solved simultaneously. There are no closed-form solutions for the equation system representing airflow and heat transfer in complicated environments such as the server rack under consideration. Therefore, computer-based numerical procedures are needed to solve this set of equations. Most commercial CFD software packages use the control volume numerical procedure for integration over the calculation domain. The integration runs into a closure problem which is resolved by introducing a turbulence model to account for different flow regimes by varying the fluid viscosity.

3.2.2 Selecting a Turbulence Model

Identifying a suitable turbulence model is very important for the accuracy of CFD simulations. ThermoStat uses the LVEL model [40], an algebraic turbulence model specifically developed for low Reynolds number flow regimes such as the ones in electronic devices. The most widely used turbulence model is the standard k-ϵ model for the wall functions in the near-wall region, but the assumption in this model of fully developed turbulent flow (high Reynolds numbers) is not applicable. The airflow in a computer rack will certainly have large regions with a low Reynolds number flow regime and, therefore, the k-ϵ model is not a suitable choice. A study [41] tested seven different turbulence models,

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

TABLE 1
CFD Simulation Parameters

Rack Parameters	
Physical Dimension (cm^3)	66 x 108 x 203
Grid Cells(#)	74 x 75 x 203
Velocity & Pressure	On
Energy Equation	Temperature Total
Turbulence Model	LVEL
Domain Material	Ideal Gas Law
Gravitational Force	On
Buoyancy Model	Boussinesq
Iterations(#)	5000
Coeff. for Auto Wall Func.	Log-law
X335 Server Sever Parameters	
Physical Dimension (cm^3)	44 x 66 x 4.4
Grid Cells (#)	55 x 80 x 15
Velocity & Pressure	On
Energy Equation	Temperature Total
Turbulence Model	LVEL
Domain Material	Ideal Gas Law
Gravitational Force	On
Buoyancy Model	Boussinesq
Iterations (#)	3500
Coeff. for Auto Wall Func.	Log-law
Outlets (#)	3

TABLE 2
Servers Inside the Rack

Geometric Information

Servers	Size (cm)			Slot number (from bottom)
	X	Y	Z	
X335 × 20	44	66	4	4-20, 26-28
X345 × 2	44	70	9	24-25,36-37
Exp300 (14 Disks)	44	52	13	38-40
Cisco Catalyst4000	44	30	27	29-34
Myrinet(M3-32P)	44	44	13	1-3

Power Information

Servers	Power (W)		Number of Components
	Min	Max	
X335 × 20	110	350	19
X345 × 2	100	660	19(10)
Exp300 (14 Disks)	280	560	22(7)
Cisco Catalyst4000	-	530	10
Myrinet(M3-32P)	-	246	9

including the standard k-ϵ model and LVEL, to find that the tested models performed better than the k-ϵ model and that LVEL, even though it is the simplest model, was as effective as the much more complicated turbulence models. This finding is very useful because significant computation time (a factor of three or higher based on the software packages and simulation settings) can be saved with the LVEL model, especially when conducting dynamic/transient CFD simulations or testing many different rack settings in steady-state conditions, as in this study.

3.2.3 CFD Tools for Computer Scientist

While researchers and students with backgrounds in mechanical engineering, thermodynamics, and fluid mechanics, are well versed in the CFD software, computer scientists and engineers have traditionally had little exposure to these tools. The graduate student(s) from computer science working on this project took around 3 months to learn this tool, with the supervision of a faculty member with expertise in CFD before we could start getting meaningful results for further fine tuning. One of the goals of ThermoStat is to facilitate easy and widespread adoption among computer scientists/engineers by hiding as many nonessential details about the CFD simulation as possible. We note that the governing equations remain the same for all different applications of airflow and heat transfer in a rack (the users need not be burdened with this information which usually requires specifying turbulence model, numerical schemes, relaxation factors, iteration settings, etc.; these values for our CFD simulation parameters are shown in Table 1), with only the boundary conditions changing for each specific rack. More specifically, the type of boundary conditions will remain the same, while the number, size, and intensity will change. For example, the dimensions and layout (which 1U slots contain servers) of a rack may be different, the number and speed of fans may change, and the power dissipation characteristics of the CPU, disk, and power supply can change. However, there are several parameters about these components that we do not need to burden the user with specifying, e.g., specifying the material parameters of components, fan configurations, etc. A user should only have to specify the dimensions of racks and server systems, the locational information on CPUs/fans/disks/power supplies, etc., and their operating power characteristics, inlet air temperature, etc. Furthermore, learning the CFD software to specify even these parameters can involve a steep learning curve. Instead, we are trying to build an XML-like configuration file specification which users can readily customize for their systems to hide all details of the CFD simulation from the user. Furthermore, we can also have default configuration files for the rack(s) that we have modeled. We believe that this approach can accelerate ThermoStat adoption, over and beyond how standard template models are being distributed for modeling electronic components with CFD software (e.g., [42], [43]), since the latter still requires learning the CFD software for using those toolboxes (Intel actually supplies a template for some of its processors for use in common CFD packages) and a sanity check needs to be done by a fluid mechanics/thermodynamics expert to ensure that the simulation is being done with the right set of parameters.

4 THERMOSTAT

4.1 CFD Model for ThermoStat

In this paper, we have enhanced our previous rack model [44] and present results for a 42U rack, with the layout of the slots in this rack given in Table 2. In this version of ThermoStat, we have modeled 20 IBM x335 servers, one IBM EXP300 storage array, and two IBM x345 management nodes on this rack. Modeling of the network (Myrinet and Cisco Gigabit Ethernet) switches is part of our future work due to the complexity of device and time constraint of simulation. With more detailed modeling of IBM x345 management nodes and IBM EXP300 storage server, the dimension of grid cell is increased from 45 $45 \times 75 \times 188$ to $74 \times 75 \times 203$ as compared to our previous version [44]. Each of the servers and storage servers is modeled as a set of components that they have. For example, each x335 server (see Table 3) has 2.8 GHz dual Xeon processors, each with a maximum power rating of 84 W when executing.

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

TABLE 3
Components Inside the x335 Server System

	Material	Heat Src.(W)
CPU[46]	Copper	31-74
Disk	Aluminum	7-28.8
Power Supply[47]	Aluminum	21-66
NIC	Copper	4
	Type	Flow Rate (m^3/sec)
Fans	Circular	0.001852 - 0.00231

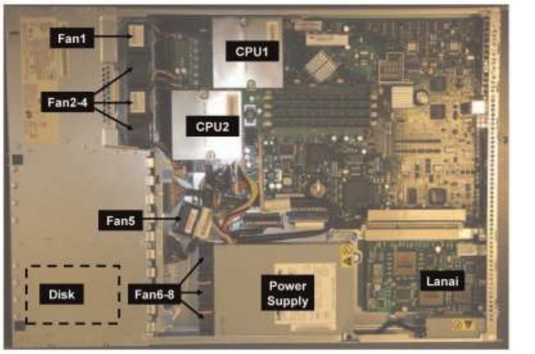


Fig. 1. Picture of the IBM X335 server. Each of the labels denotes the names of major components. Note that the hard disk is hidden by a cover on the front side of the server.

However, the data sheets for the processor suggest using a maximum value of 74 W, which is the Thermal Design Power (TDP), for thermal modeling. When the CPU is idling, we assume an idle power of 31 W (measured values from [45]). We modeled components that are only important and directly relevant to the thermal behavior of system. The number of components modeled for each server is shown in Table 2, where the numbers in parentheses are those of components used in the previous experiment [44]. The accuracy of simulation depends on the simulated components for each server. We divided the front (inlets) area of the rack into eight vertical regions and used measured values of the inlet air temperature for these servers, as shown in Table 4 (the higher numbers are on top). Note that more accurate power values based on detailed modeling/information and/or measurements can be used as well. Furthermore, the processor on our system does not allow any frequency/DVS capabilities. For some of the later experiments in this paper, when assuming frequency modulation abilities, we use a simple linear dependence model between frequency and power consumption (without any voltage changes) for illustration purposes. Each x335 server has an SCSI disk, Myrinet NIC, eight fans, and a power supply, whose layout is given in Fig. 1, and the associated modeling parameters are given in Table 3. The eight circular fans direct most of the air flow in the server, taking in the air through vents at the front of the case and directing it out to the vents at the back. In addition, there is an inlet at the inside base (behind the machines) of the rack which brings in air flow from the raised floor. Wires and guiding components at the back of the rack are not being modeled for simplicity and we found that these do not significantly impact the temperature within each server system. The number of grid cells and iteration counts for running the simulations have been set after experimentally determining trade-offs between speed and accuracy.

TABLE 4
Front Inlet Temperature Distribution

Inlet Temperature								
Location	1	2	3	4	5	6	7	8
Temp. (C)	15.3	16.1	18.7	22.2	23.9	24.6	25.2	26.1

Most academic institutions have licenses (running to a couple of thousand dollars) for popularly used CFD software such as FLUENT, FLOTHERM, and Phoenics. We are currently using Phoenics [48] (which was, in the past, distributed as free Shareware) for ThermoStat due to its simple interface, which enables users to employ only Cartesian coordinates. More advanced software with body-fitted coordinates gives significant advantages for curvilinear systems, but its simulation domain layout settings require much more intensive preprocessing that is not really useful for simulating rack-mounted systems.

4.2 Validation of the Server Model

To validate our ThermoStat model, we deployed 29 temperature sensors (DS18B20 from Dallas Semiconductor [49]) at different points in our rack-mounted system, both within the individual x335 server systems and at the rear (inside) of the rack whose temperatures are affected by the individual server systems, and compared those readings with the predicted temperatures by our ThermoStat model at those points. The DS18B20 sensor has an accuracy of $\pm 0.5^\circ\mathrm{C}$ accuracy from a range of $-10^\circ\mathrm{C}$ to $+85^\circ\mathrm{C}$. We read the temperature by using 1-wire interface and it gives a 12-bit reading within 750 ms. Fig. 2a shows the placement of 11 sensors within the server system. Note that not all sensors are on the surface of the components and some of them are suspended in the air from the roof of the case. Two of the sensors—10 and 11—were stuck to the surfaces of the disk and CPU 1, respectively, with a thermal paste. In the case of sensor 11, we could not stick it directly to the CPU surface because of the heat sink: We did not want to run the system after removing the heat sink due to fear of damaging it. We could not stick it to the base at the center of the heat sink because the sensor was not small enough to fit between the fins. Instead, it was stuck to the side at the base of the heat sink, where the temperatures are expected to be lower than those at the center of the CPU surface. As noted in [50], there can be as much as a $10^\circ\mathrm{C}$ difference in temperatures across the chip. In fact, our ThermoStat model gives the CPU surface center temperature $38^\circ\mathrm{C}$ in the idle state, which fell in this range when the reading at sensor 11 was around $29^\circ\mathrm{C}$. We are currently trying to fix smaller sensors between the fins for more accurate validation of the CPU center temperature. We show the validation results when the components are idle (i.e., CPUs, disks, power supplies, and fans are operating at the lower end of their power range specified in Table 1) in Fig. 3b.

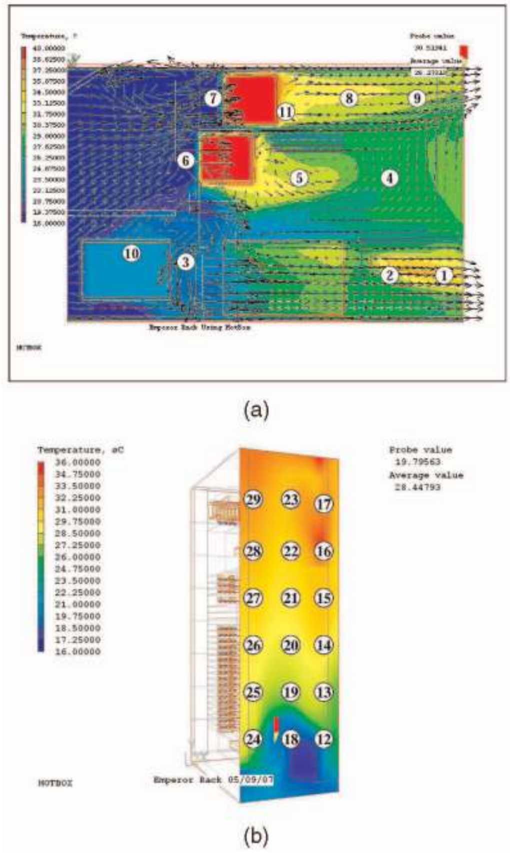


Fig. 2. Validation: Sensor placement locations (a) within the X355 server and (b) at the back of rack. Note that the color coding for the temperature is for a cross section of the shown spatial extent and does not necessarily reflect the surface temperatures of components.

When we examine the results within the server system, we notice that our model closely (2-3°C) follows the sensor measurements. Across all 11 sampled points, the average absolute error is around 9 percent. We note that we are getting close agreement, despite the following discrepancies that can arise:

- The manufacturer rates these sensors with an error margin of ±0.5°C. Further, even though these sensors are fairly small/thin, they are still not measuring the temperature at a single point in space.
- Even though we took great care to position the sensors (and measure these positions) and not move these positions when closing the cases/doors, there are still bound to be some errors/distortions in the spatial locations of where we are measuring the data. As the temperature profile in Fig. 2a for a vertical cross section of a particular spatial region shows (from the ThermoStat model), there can be substantial changes in temperature by moving the sensors by even 1 or 2 cm.

Note that we could not insert sensors into the CPU's heat sink (where it attaches to the CPU). Similarly, we could not put sensors into disks. Instead, in these two components, we relied on the sensors already built into the hardware, which we read from in software. We compared these readings with those predicted by our ThermoStat model as well and, as we can find, we are very close in the case of the CPU (less than 2 percent error), where the location of the sensor on the Xeon is well documented, while, in the case of the Disk (whose location is not well documented), there is a 12 percent error.

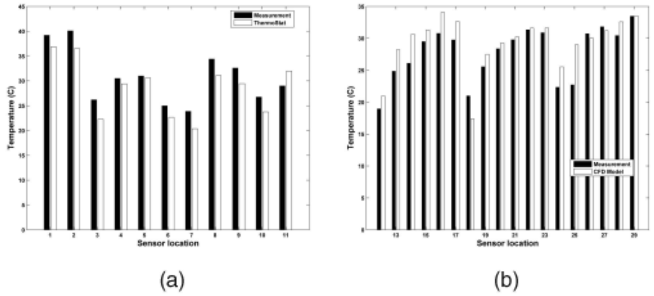


Fig. 3. Validation: Comparing temperature from CFD modeling and sensor measurements. (a) Within the server. (b) Back of the rack.

4.3 Validation for Rack

Fig. 2b shows the sensor placement location of 18 sensors to the back door of the rack. All sensors were stuck to the back door of the rack. To minimize the measurement error, we use adhesives to stick the sensor on the door to prevent it from unnecessary movement. We improve our model's simulation accuracy by additionally modeling components within the servers. As a result, the average error margin decreases from 11.00 percent to 7.85 percent as compared to our previous model [44], as shown in Fig. 3b. However, the simulation time increases by 75 percent. Errors across the locations of a rack are almost evenly distributed, except for a few points (such as sensors 18, 24, and 25). This is because we have ignored wires around the inlet on a computer room floor. This inlet is located inside the rack and it delivers cooling air to the back of the rack. It is covered with a bunch of wires to supply power to the rack. Due to the irregularity of its shape, it is hard to model these wires on the ThermoStat model.

In addition to these sensor measurements, we also took a thermal image by using an infrared camera at the back of the rack (surface temperature) and we found that the thermal profiles are quite close to that predicted by the ThermoStat model. First, this statement is supported by the similarities of graphical patterns in the two images shown in Fig. 4. This indicates that the temperature distribution produced by the ThermoStat model matches the actual distribution captured by the IR camera. Second, the temperature range obtained by the ThermoStat model also falls close to the measured temperature range by the IR camera. The measured temperature distribution ranges from 21.7°C to 35.0°C and the ThermoStat model's temperature varies from 17.9°C to 33.57°C. In terms of the maximum and minimum values, the error margin is 4.0 percent and 17.5 percent, respectively.

When evaluating these errors, it is important to note that these CFD simulations involve complex physical phenomena of combined airflow with conductive, convective, and radiative heat transfer. These kinds of CFD simulations were only made possible in the last 10 years due to the rapid

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

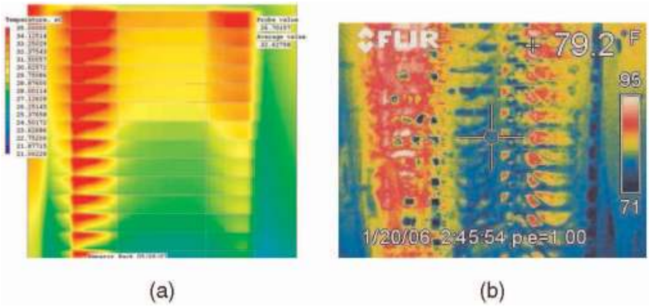


Fig. 4. Validation: Temperature distribution with (a) the ThermoStat and (b) IR camera pictures. Note that temperature in (b) is in degrees Fahrenheit.

development of computer power. Typically accepted errors by CFD modelers when evaluating combined airflow with heat transfer in realistic environments are around 20 percent [51] in regions with separation flows, such as rack simulations by ThermoStat. A recent CFD study [52] evaluated the performance of different turbulence models for electronic applications revealed errors greater than 20 percent for different simulation parameters. Therefore, the ThermoStat model is capable of correctly predicting temperatures in computer racks and will be further developed as CFD models for electronic applications are improved.

5 Thermal Study on the Rackmount Server

In this section, we utilize ThermoStat for static machine design to study thermal behavior in a server system and an entire rack. Thermal interaction between components such as a processor and a disk is exploited and thermal dependency between servers within the rack is studied.

ThermoStat is supported on any platform running Linux. Phoenics should be on a separate system. Also, an XML program is needed to create XML files which should be input to ThermoStat. They should be created for the server and rack-mounted servers. There are many distributions for XML programs. Also, we can use a normal text editor. Before running ThermoStat, we need to prepare system input parameters for the rack/server system that we are modeling. Detailed usage and reference about how we use ThermoStat can be found on our ThermoStat Web site [53].

5.1 Study on the Characteristics of the Existing System

5.1.1 Are Servers in a Rack Independent?

It is interesting to see how machines in a rack influence each other's temperature, if at all. In our modeled rack, air flows in through the front of the machines, drawn in by fans, and exits at the rear. The rear is thus hotter than the front and, as expected, it is hotter nearer the top. We picked four machines—1, 5, 15, and 20 (in increasing order from the bottom of rack)—for comparing the thermal profiles. All of these machines are in the idle mode. Fig. 5 compares the spatial temperature difference between pairs of these machines. As we can see, machines at the top are hotter than those below, with around 7-10°C difference in temperature between machines 20 and 1. The magnitude of this difference decreases with less distance between the machines, as can be seen in Fig. 5b, where machines 15 and 5 differ by 5-7°C. Such information can be useful for performing temperature aware scheduling and load management, e.g., assigning a higher load to machines at the bottom of the rack. From all of these graphs, we observe that, even though there are deviations from zero (i.e., there is a temperature difference between the two profiles being compared), the differences are very minor, going to at most 2-3°C difference in the temperatures. This suggests that machines are relatively insulated from each other and are not significantly affected by the load imposed on the other machines in the studied rack. Note that this is probably a consequence of how the air flow has been designed by the engineers in the first place. Still, these results are useful not only for engineers in packaging systems but also in designing ThermoStat itself (if initial studies suggest insulation between machines, we could focus on more detailed studies on individual server systems rather than study the entire rack) for lowering simulation time.

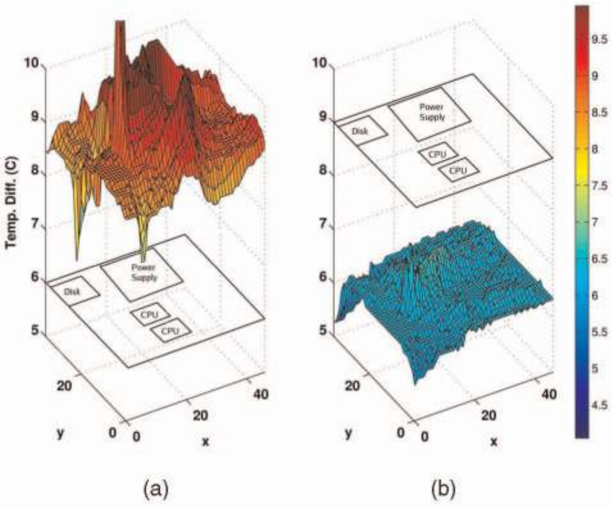


Fig. 5. Temperature difference between servers of a rack. The locations of major components in the server are projected in each graph.

5.1.2 Are Components in a Server Independent?

Static design considerations when packaging components within a server system include understanding 1) the range of inlet temperatures for safe operation of components, 2) whether the provisioned fans are able to adequately cool the components, and 3) how the heat generated by components interacts with the heat generated by the other components (i.e., are they laid out properly). In the previous section, we already studied issues related to inlet temperature and fan operation and, in Fig. 6, we examine how components affect each other's temperatures, if at all. In these experiments, for each computational component—CPUs 1 and 2 and the Disk—we consider two possibilities: whether they are idle (consuming much lower power) or whether they are operating at maximum power. In addition to the temperatures of individual components, the graph also plots the average temperature within the server system. As the results in this graph show, even though the average temperature of the spatial extent changes with the load on the components, the components exhibit little interaction between each other on the modeled system. This is due to the design of the x335 server, where the components are laid out fairly well apart (see Fig. 1) and

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

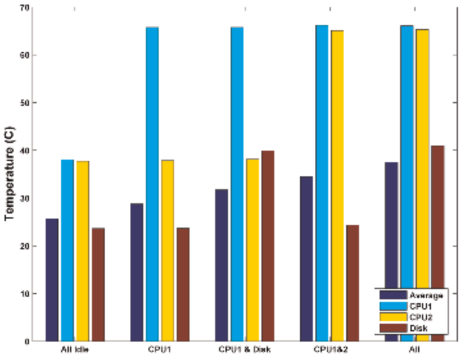


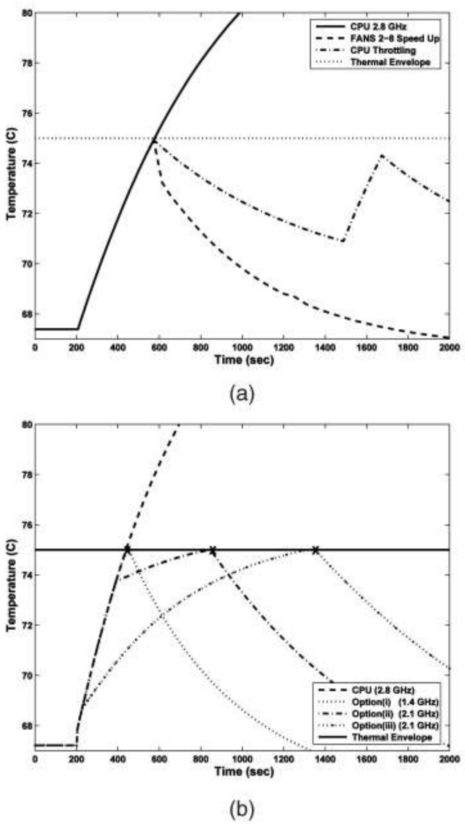
Fig. 6. Examining interactions, if any, between components. Legends on the x-axis indicate which components are active (running at maximum power), with the rest being idle.

the fans are placed and directed so that the hot air from one component does not really blow over the others studied here. The engineers have done such studies when laying out the components and provisioning the cooling systems. Note that we have already shown in the previous section that the component temperatures are significantly impacted by the fans, i.e., the air flow directed by them, and one should not misunderstand the results in Fig. 6 to imply that each component's temperature is dependent only on its own characteristics (power, materials, etc.).

Studying temperature interactions is very important and, until now, it has been mainly packaging engineers who have been studying these issues with their own proprietary tools. ThermoStat opens the opportunity for computer architects and systems researchers to study these issues as well. Leaving it entirely to packaging engineers and cooling systems can unduly increase the cost. We are already seeing sophisticated layouts and airflow techniques in dense blade servers. For instance, in IBM's HS20 blade server [54], the two CPUs occupy nearly 1/3 of the floor area, making it very difficult to avoid having the air flow from one to the other. The air inlet is not in the front for this system and is near a memory bank instead. Furthermore, the designers also pulled out the power supply from within this blade server, using a centralized supply to power several blades. A sophisticated vertical air flow through the circuit boards is also being used on the dense BlueGene/L system. With growing densities in integration at the complete system level, the importance of high-level optimizations, rather than just packaging, become more important. This is akin to how microarchitectural management of temperature is becoming important over and beyond packaging optimizations.

6 Designing DTM Techniques at a Rack-Mounted Server

ThermoStat can also be used for designing and evaluating DTM techniques. In this section, we first illustrate this below with two examples to show how ThermoStat can help design both reactive and proactive DTM techniques for controlling CPU temperature. Then, we propose an isothermal workload distribution mechanism that can leverage temperature around the rack for the given power budget and work requirement.



(a)

(b)

Fig. 7. Designing DTM Techniques with ThermoStat. (a) Fan 1 fails at 200 s. (b) Inlet air temperature suddenly changes from $18^\circ\mathrm{C}$ to $40^\circ\mathrm{C}$ at 200 s.

6.1 What Should We Do When a Fan Breaks: A Reactive Example

In this example, we make fan 1 break down at time 200 s (see Fig. 7a), causing CPU 1's temperature to start rising rapidly. The thermal envelope of safe CPU operation is set to $75^\circ\mathrm{C}$ [46] and, if there is no management technique, ThermoStat shows us that the CPU temperature running at 2.8 GHz will exceed this thermal envelope 370 seconds after this event. Note that just using sensors on the actual system may not give this predictive information, i.e., whether the temperature will exceed the envelope and, if so, at what time. Allowing the CPU to operate as is beyond this point is not safe and ThermoStat can help us evaluate which remedial/reactive measure should be taken to control its temperature. In Fig. 7a, we consider two possible reactive measures when reaching this threshold. The first option is to make all of the other fans 2-8 spin faster. Note that the fans in our system allow multiple speeds of operation. In the default operation, their CFM is 0.00185 m^3/s and we change this to 0.00231 m^3/s. As we can see, this compensates for any rise in temperature, which is, again, a piece of information that would not be available without modeling air flow. The other reactive measure that we consider is cutting down the CPU's operating frequency by 25 percent, i.e., it now runs at 2.1 GHz, which is also effective for cooling down the CPU. This would be an option only on processors capable of such control (which is becoming quite prevalent). It is also possible that, once the CPU cools sufficiently, its speed could again be ramped up (as shown at around 1,500 seconds) and so on. Between these two options, the former

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

may be preferable if performance is more critical since this option does not lose any CPU capacity. In this example, ThermoStat helps us identify the possible reactive options, evaluate their effectiveness, and quantify the times to get to these associated temperatures.

6.2 What Should We Do When the Inlet Air Temperature Suddenly Rises: A Proactive Example

In this example, we make the inlet air temperature suddenly go up to $40^\circ\mathrm{C}$ from $18^\circ\mathrm{C}$ at 200 s, as shown in Fig. 7b. Though such an instantaneous change is somewhat drastic (i.e., machine room temperatures vary due to CRAC breakdown, doors left open, sudden load surges, etc.), we are using this example for illustrative purposes. ThermoStat shows that the temperature will reach the envelope in another 220 seconds in this case.

Rather than waiting until reaching the thermal envelope, at which point we may have waited too long, one may want to take a more proactive thermal management strategy, i.e., take remedial actions before the emergency point. At the same time, taking the actions too early, say, immediately after noting the inlet air temperature change, may be too conservative and can lower performance (if the DTM technique scales back the frequency). We wish to mention that, under the $40^\circ\mathrm{C}$ operating conditions, scaling back the CPU frequency by 25 percent does not really keep the temperature within the envelope and we use a 50 percent scaled frequency value to keep the temperature within bounds.

In this example, we show three possible thermal management options to not exceed the thermal envelope: 1) running the CPU at full frequency until the emergency point, at which point $(\mathrm{time} = 440\ \mathrm{s})$, scaling back the frequency by 50 percent, which is what the purely reactive approach would do, 2) running the CPU at full frequency for another 190 s after detecting inlet air temperature change, then (at $\mathrm{time} = 390\ \mathrm{s}$) resorting to a 25 percent frequency scale back, and later, when reaching emergency (at $\mathrm{time} = 821\ \mathrm{s}$), cutting the CPU frequency further to 50 percent of maximum value, and 3) running the CPU at full frequency for another 28 s, then (at $\mathrm{time} = 228\ \mathrm{s}$) scaling back the CPU by 25 percent and scaling back the CPU to 50 percent when reaching the emergency threshold at 1,317 s. The choice of which option should be used depends on the workload. For instance, if the amount of work remaining to be done requires 500 s when operating at full speed, the three options would complete this job at 960, 803, and 857 s, respectively, making option 2 preferable in this example.

Even though we have shown only CPU throttling in this example for temperature management, there could be scenarios where a combination of different techniques (e.g., throttling + fan control) could be exploited using the ThermoStat infrastructure.

6.3 Isothermal Workload Distribution

ThermoStat can be a useful tool for data center management. It can be used to control the temperature of specific racks based on the cooling budget in the data center or to efficiently distribute the workload to the servers of the rack to attain uniform temperature distribution. We illustrate two knobs to efficiently manage these requirements with ThermoStat. With extensive simulations using ThermoStat, we found that the two primary driving factors in determining the CPU temperature are 1) the load (power expended by its CPUs running at a given frequency) on that server and 2) the position (slot number in Table 2) of that server in the rack from the bottom (since the inlet air temperature to the server is different accordingly). These two factors are much more important in determining the server temperature than the load imposed on the adjacent servers. To illustrate this, we show the results of temperature difference from the following three load conditions on the rack:

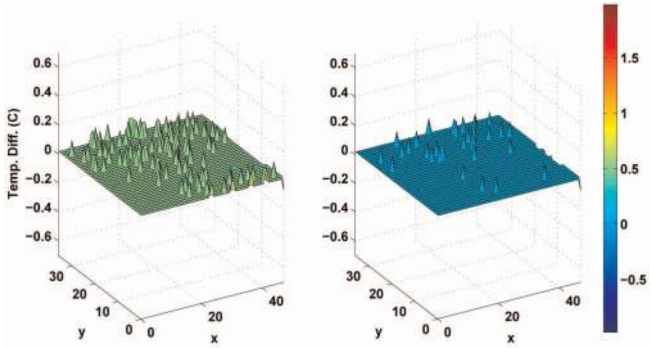


Fig. 8. Temperature difference for the same server (i.e., server 10) on adjacent servers' different workloads.

- *Load 1.* Every server of the rack runs at the same frequency (1.75 GHz).
- *Load 2.* One-third of the rack from the bottom runs at 1.4 GHz, one-third of the rack from the top runs at 2.1 GHz, and the rest of the rack runs at 1.75 GHz.
- *Load 3.* Servers run at 1.575, 1.75, and 1.925 GHz for three different regions of the rack (bottom, middle, and top).

Fig. 8 compares the spatial temperature difference of server 10 (positioned at the middle of the rack and running at 1.75 GHz over all simulations). It clearly shows that the temperature of server 10 is not affected by the load on adjacent servers. This observation helps us create a database of $< server\ id,\ load/CPU\ frequency,\ Temperature >$ for each server on the rack, independent of the other servers. By preconstructing such a database, we can answer the following two conditions.

6.3.1 What Is the Maximum Workload That Can Be Sustained with a Given Threshold Temperature Bound for Each Server in the Rack?

In this case, we have a mission to manage the data center to attain a safe operating temperature for any rack at the data center. We want to maximize the throughput of a rack while the rack is still running under the given temperature threshold. In Fig. 9, we show results from two workload placement policies. In both experiments, we attain the peak rack temperature below a given threshold temperature bound:

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

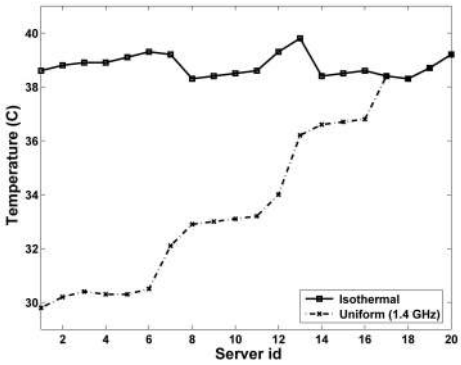


Fig. 9. Isothermal workload distribution for a given temperature. For a fixed temperature of $40^{\circ}\mathrm{C}$, the isothermal workload can be defined as (server id, frequency for isothermal) = {(1-5, 2.275 GHz), (6, 2.1 GHz), (7-12, 1.925 GHz), (13, 1.75 GHz), (14-16, 1.575 GHz), (17-20, 1.4 GHz)}.

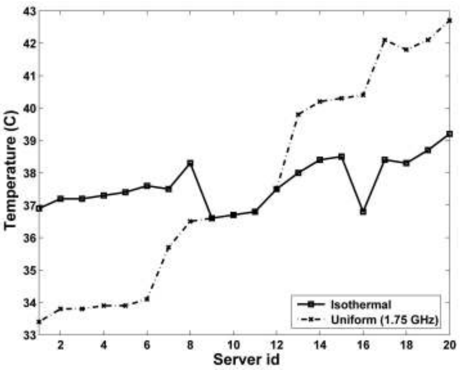


Fig. 10. Isothermal workload distribution for a given work request. The work request is given as an average aggregated frequency of 1.75 GHz. The isothermal workload can be defined as (server id, frequency for isothermal) = {(1-6, 2.1 GHz), (7-8, 1.925 GHz), (9-12, 1.75 GHz), (13-15, 1.575 GHz), (16-20, 1.4 GHz)}.

- *Uniform load.* This is the normal load balancing technique, where the given total load to the rack is evenly split across the servers as long as no server exceeds a temperature threshold (in this example, it is set to $40^{\circ}\mathrm{C}$). As expected, this threshold can be reached for the servers at the top of the rack (to the right on the x-axis), whereas below can accommodate a higher load without exceeding a given temperature bound.
- *Isothermal load.* This is the result of using an algorithm (shown in Algorithm 1) that can exploit the ThermoStat provided temperature database to ensure that servers lower on the rack are, in fact, assigned a higher workload, which still keeps the temperature within the $40^{\circ}\mathrm{C}$ bound.

Algorithm 1: fixed temperature threshold.

```
Step 1. Define the array of servers at the rack-mounted
server:
    $U = \{U_i = (f, t) \mid f : frequency,\ t : temperatureN\}$
        Where i = 1 to $N$
Step 2. Set the frequency of each server with its
    temperature below a given threshold ($T_{threshold}$):
  $M$: total number of frequency scaling steps
    $for(i = 1\ to\ N)\{$
      $for(j = 1\ to\ M)\{$
        $if((t = ThermoStat_{Search}(i, j)) > T_{threshold})\{$
          $U_i = (j, t)$
          $break$
        $\}$
      $\}$
    $\}$
```

The result of our ThermoStat benefited isothermal strategy in this case can accommodate 32.5 percent higher workload than the uniform workload allocation strategy. This amount of improvement in the throughput can reduce the cooling cost of the data center.

6.3.2 For a Given Aggregate Throughput, How Do We Allocate the Load to Each Server to Reduce the Maximum Temperature on Any Server in the Rack?

In this scenario, we have a given workload requirement to be carried out. However, we do not want to get high raised peak temperature around the rack. By leveraging the temperature of individual servers, we can report lower rack temperature to the data center thermal management system. Based on each rack temperature, the data center management system will decide on the operating status of the cooling equipment in the data center. Hence, the temperature of a single rack can affect the cooling cost of the overall data center. In Fig. 10, we present the results from two experiments. In these experiments, we have a given work request as a form of the aggregate throughput. To simplify the problem, we assume that the throughput of the processor is proportional to the frequency of the processor.

- *Uniform Load.* We evenly balance the given load to each of the servers. As we can see, the temperature can vary from around $33.5^{\circ}\mathrm{C}$ on a server at the bottom to as high as $42.5^{\circ}\mathrm{C}$ on a server at the top.
- *Isothermal Load.* We attempt to assign the given load to balance the temperature across the servers as described at Algorithm 2. This algorithm ensure that peak temperature around the rack is minimized while the specified throughput is attained by servers at rack.

As we can see, the isothermal workload assignment policy can meet the same throughput requirements of the composed load with a much lower temperature variance across the servers. The peak temperature of the rack is lowered by $3.5^{\circ}\mathrm{C}$. This would be preferred when the higher temperature on one or more servers is highly undesirable from the reliability perspective.

Algorithm 2: fixed total power budget.

```
Step 1. Define the array of servers at the rack-mounted
server:
        $U = \{U_i = (f, t) \mid f : frequency,\ t : temperature\}$
            Where i = $1, \ldots, N$
Step 2. Uniformly allocate the frequency to every server
        and search for the current frequency and
```

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

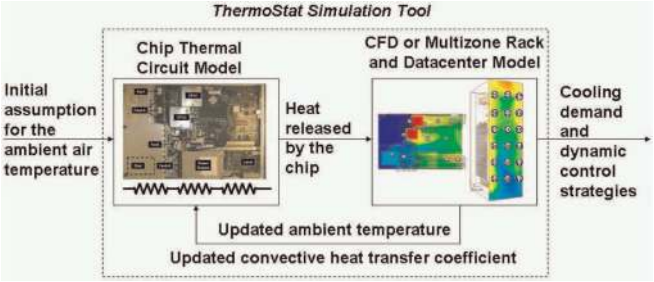

ThermoStat Simulation Tool
Chip Thermal Circuit Model
CFD or Multizone Rack and Datacenter Model
Initial assumption for the ambient air temperature
Heat released by the chip
Cooling demand and dynamic control strategies
Updated ambient temperature
Updated convective heat transfer coefficient

Fig. 11. Integration of airflow and thermal models for chip and ThermoStat.

```
        temperature of each server from ThermoStat:
        $F(U_i)$: frequency for the $i$th server
        $T(U_i)$: temperature for the $i$th server
            $for\ i = 1, \ldots, N$
                $F(U_i) = \frac{F(U)}{N}$
                $T(U_i) = ThermoStat_{Search}(i, F(U_i))$
Step 3. Distribute the frequencies for isothermal workloads:
        Sort the servers by temperature, $u = Temp_{Sort}(U)$
        Find the servers with maximum and minimum
        temperatures,
            $u_{max} = Temp_{Max}(U)$, $u_{min} = Temp_{Min}(U)$
        Balance the frequencies across servers,
            $while(|T(U_{max}) - T(U_{min})| < \delta)\{$
                $u_{max} = Temp_{Max}(U)$, $u_{min} = Temp_{Min}(U)$
                $Freq_{down}(U_{max})$, $Freq_{up}(U_{min})$
            $\}$
```

6.4 Integration between ThermoStat and the Component-Level Model

The chip model and ThermoStat have to be integrated to enable engineers to perform optimization of these integrated systems. At present, these models are typically decoupled because the information needed for the model coupling requires research integrating computer architecture with cooling strategies. The computer-architecture-oriented approach is focused on the thermal circuit modeling of microprocessors [55], while the cooling system strategies use CFD modeling to design the layout of the data center space under steady-state conditions [56]. Integration, especially under the dynamic conditions, is still a big challenge, but is a promising research area for enabling better computer rack performance while saving energy. Fig. 11 outlines our proposed integration of the chip, rack, and data center models. To determine the power output of the chip, the thermal circuit model requires information on the ambient air temperature and convective heat transfer coefficient, which are directly affected by the data center cooling system and computer rack architecture. Therefore, an iterative procedure is proposed to update both the models, the chip model with realistic ambient temperature and convective heat transfer coefficient, and the ThermoStat with the realistic power/heat output from the chip and other heat sources in a computer rack.

7 Conclusions and Future Work

This paper has presented a CFD-based tool, called ThermoStat, for obtaining thermal profiles of rack-mounted servers. ThermoStat can be used in both system building/packaging studies to figure out how we can place components, design cooling systems, etc., and to undertake higher level (architectural/software) thermal optimization studies. Until now, such tools have been mainly restricted to the industry, and ThermoStat is intended to fill this void in the research/academic community. We have released ThermoStat for public download. Usage of this tool is not expected to require extensive knowledge of CFD since we are abstracting most of the interactions with the underlying simulation engine by an easy-to-use XML-like interface. It is currently implemented on the Phoenics CFD software and future work can look at adapting it for other popular (and public domain) CFD engines.

We have also presented that ThermoStat can be used for undertaking higher level (architectural/software) thermal optimization studies. We envision a database of parameterized options built using ThermoStat in an offline fashion for different system events and operating conditions, which can then be consulted at runtime for decision making. The number of events (e.g., fan failures and inlet temperatures) is not expected to be excessively high. We have shown that ThermoStat helps us to identify the possible reactive and proactive options, evaluate their effectiveness, and quantify times to get to these associated temperatures. Finally, we illustrated a workload placement mechanism at a rack by using ThermoStat. The isothermal workload distribution strategy that benefited in this case can accommodate 32.5 percent higher workload than the uniform workload allocation strategy, which still keeps all server within the specified temperature bound. For a given workload, we can reduce the peak temperature around the rack by $3.5^\circ\mathrm{C}$ by using the isothermal workload distribution strategy.

We can extend our research to develop a holistic thermal management scheme, which determines power/thermal control of system dynamically based on different operating conditions and integrates the computer systems and cooling system for decision making. These thermal determinations and enforcement are done in a holistic way, from the chip level to the data center level, across different layers of the hardware and software stack. An integrated control loop that spans both computing and cooling systems is needed to dynamically determine thermal control for cost-effective and reliable operation. Decisions such as is it better to switch off servers as opposed to blowing colder air can be made based on which is more cost-effective at that point in time. A holistic thermal/power management approach is needed for designing such a controller, which considers the issues from the individual chips to the entire data center. This paper can be a good starting point for these research. We also plan to develop interfaces between this framework and other architectural performance/power/thermal modeling tools being used by the community.

References

[1] S. Gurumurthi, A. Sivasubramaniam, and V. Natarajan, "Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management," *Proc. 32nd IEEE Int'l Symp. Computer Architecture,* pp. 38-49, June 2005.

[2] S. Charrap, P. Lu, and Y. He, "Thermal Stability of Recorded Information at High Densities," *IEEE Trans. Magnetics,* vol. 33, no. 1, pp. 978-983, Jan. 1997.

[3] L. Yeh and R. Chy, *Thermal Management of Microelectronic Equipment.* Am. Soc. of Mechanical Eng., 2001.

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

[4] S. Gunther, F. Binns, D. Carmean, and J. Hall, "Managing the Impact of Increasing Microprocessor Power Consumption," *Intel Technology J.,* vol. 5, Feb. 2001.

[5] H.F. Hamann, A. Weger, J. Lacey, Z. Hu, P. Bose, E. Cohen, and J. Wakil, "Hotspot-Limited Microprocessors: Direct Temperature and Power Distribution Measurements," *IEEE J. Solid-State Circuits,* vol. 42, pp. 56-65, Jan. 2007.

[6] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-Aware Microarchitecture," *Proc. 30th IEEE Int'l Symp. Computer Architecture,* pp. 1-13, June 2003.

[7] A. Weissel and F. Bellosa, "Dynamic Thermal Management for Distributed Systems," *Proc. First Workshop Temperature-Aware Computer Systems,* June 2004.

[8] Y. Li, K. Skadron, Z. Hu, and D. Brooks, "Performance, Energy, and Thermal Considerations for SMT and CMP Architectures," *Proc. 11th Int'l Symp. High-Performance Computer Architecture,* pp. 71-82, Feb. 2005.

[9] J. Srinivasan and S. Adve, "Predictive Dynamic Thermal Management for Multimedia Applications," *Proc. 17th Int'l Conf. Supercomputing,* pp. 109-120, June 2003.

[10] J. Hasan, A. Jalote, T.N. Vijaykumar, and C. Brodle, "Heat Stroke: Power-Density-Based Denial of Service in SMT," *Proc. 11th Int'l Symp. High-Performance Computer Architecture,* pp. 166-177, 2005.

[11] Y. Kim, S. Gurumurthi, and A. Sivasubramaniam, "Understanding the Performance-Temperature Interactions in Disk I/O of Server Workloads," *Proc. 12th Int'l Symp. High-Performance Computer Architecture,* pp. 179-189, Feb. 2006.

[12] T. Heath, A.P. Centeno, P. George, Y. Jaluria, and R. Bianchini, "Mercury and Freon: Temperature Emulation and Management in Server Systems," *Proc. 12th Int'l Conf. Architectural Support for Programming Languages and Operating Systems,* Oct. 2006.

[13] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," *Proc. 27th Int'l Symp. Computer Architecture,* pp. 83-94, June 2000.

[14] W. Chen, M. Dubois, and P. Stenstrom, "Integrating Complete-System and User-level Performance/Power Simulators: The SimWattch Approach," *Proc. IEEE Int'l Symp. Performance Analysis of Systems and Software,* 2003.

[15] D. Kudithipudi, S. Petko, and E. John, "Cache Leakage Power Analysis in Embedded Applications," *Proc. 47th IEEE Midwest Symp. Circuits and Systems,* pp. II517-II520, 2004.

[16] R. Bianchini and R. Rajamony, "Power and Energy Management for Server Systems," *Computer,* vol. 37, no. 11, Nov. 2004.

[17] J.S. Chase, D.C. Anderson, P.N. Thakar, A.M. Vahdat, and R.P. Doyle, "Managing Energy and Server Resources in Hosting Centers," *Proc. 18th ACM Symp. Operating System Principles,* Oct. 2001.

[18] F. Bellosa, S. Kellner, M. Waitz, and A. Weissel, "Event-Driven Energy Accounting of Dynamic Thermal Management," *Proc. Workshop Compilers and Operating Systems for Low Power,* Sept. 2003.

[19] R. Huang and D. Chung, "Thermal Design of a Disk-Array System," *Proc. Eighth InterSociety Conf. Thermal and Thermomechanical Phenomena in Electronic Systems,* pp. 106-112, May 2002.

[20] D. Brooks and M. Martonosi, "Dynamic Thermal Management for High-Performance Microprocessors," *Proc. Seventh Int'l Symp. High-Performance Computer Architecture,* pp. 171-182, Jan. 2001.

[21] M. Gomaa, M.D. Powel, and T.N. Vijaykumar, "Heat-and-Run: Leveraging SMT and CMP to Manage Power Density through the Operating System," *Proc. 11th Int'l Conf. Architectural Support for Programming Languages and Operating Systems,* pp. 260-270, 2004.

[22] L. Shang, L.-S. Peh, A. Kumar, and N. Jha, "Thermal Modeling, Characterization and Management of On-Chip Networks," *Proc. 37th IEEE Int'l Symp. Microarchitecture,* pp. 67-78, Dec. 2004.

[23] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "DRPM: Dynamic Speed Control for Power Management in Server Class Disks," *Proc. 30th IEEE Int'l Symp. Computer Architecture,* pp. 169-179, June 2003.

[24] E. Pinheiro and R. Bianchini, "Energy Conservation Techniques for Disk Array-Based Servers," *Proc. 18th IEEE Int'l Conf. Supercomputing (ICS '04),* June 2004.

[25] Q. Zhu, F. David, C. Devraj, Z. Li, Y. Zhou, and P. Cao, "Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management," *Proc. 10th Int'l Symp. High-Performance Computer Architecture,* 2004.

[26] N. Rasmussen, "Guidelines for Specification of Data Center Power Density," APC white paper, 2004.

[27] R.K. Sharma, C.E. Bash, and C.D. Patel, "Dimensionless Parameters for Evaluation of Thermal Design and Performance of Large-Scale Data Centers," *Proc. Eighth ASME/AIAA Joint Thermophysics and Heat Transfer Conf.,* June 2002.

[28] C.E. Bash, C.D. Patel, and P.K. Sharma, "Efficient Thermal Management of Data Center: Immediate and Long-Term Research Needs," *Int'l J. Heat, Ventilating, Air-Conditioning and Refrigeration Research Needs,* vol. 9, no. 2, pp. 137-152, 2003.

[29] P. Rodgers and V. Eveloy, "Prediction of Microelectronics Thermal Behavior in Electronic Equipment: Status, Challenges and Future Requirements," *Proc. Ninth Int'l Conf. Thermal and Mechanical Simulation and Experiments in Micro-Electronics and Micro-Systems,* 2003.

[30] J.F. Karlsson and B. Moshfegh, "Investigation of Indoor Climate and Power Usage in a Data Center," *Energy and Buildings,* vol. 37, pp. 1075-1083, 2005.

[31] C.D. Patel, R. Sharma, C.E. Bash, and A. Beitelmal, "Thermal Considerations in Cooling Large-Scale High Compute Density Data Centers," *Proc. Eighth Intersoc. Conf. Thermal and Thermomechanical Phenomena in Electronic Systems,* May 2002.

[32] C.D. Patel, C.E. Bash, C. Belady, L. Stahl, and D. Sullivan, "Computational Fluid Dynamics Modeling of High Compute Density Data Centers to Assure System Inlet Air Specifications," *Proc. Pacific RIM/ASME Int'l Electronic Packaging Technical Conf. and Exhibition,* July 2001.

[33] C.D. Patel, C.E. Bash, and M. Beitelmal, "Smart Cooling of Data Centers," *Proc. Pacific RIM/ASME Int'l Electronics Packaging Technical Conf. and Exhibition,* July 2003.

[34] M.H. Beitelmal and C.D. Patel, "Thermo-Fluids Provisioning of a High Performance High Density Data Center," Technical Report HPL 2004-146 (R.1), Hewlett-Packard Laboratories, 2004.

[35] C.D. Patel and A.J. Shah, "Cost Model for Planning, Development and Operation of a Data Center," Technical Report HPL 2005-107 (R.1), Hewlett-Packard Laboratories, 2005.

[36] S.V. Patankar and K.C. Karki, "Distribution of Cooling Airflow in a Raised-Floor Data Center," *Proc. Am. Soc. Heating, Refrigerating, and Air-Conditioning Eng.,* 2004.

[37] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making Scheduling 'Cool': Temperature-Aware Workload Placement in Data Centers," *Proc. Usenix Ann. Technical Conf.,* Apr. 2005.

[38] J. Moore, R. Sharma, R. Shih, J. Chase, C.D. Patel, and P. Ranganathan, "Going Beyond CPUs: The Potential of Temperature-Aware Solutions for the Data Center," *Proc. First Workshop Temperature-Aware Computer Systems* , May 2002.

[39] R. Sharma, C. Bash, C. Patel, R. Friedrich, and J. Chase, "Balance of Power: Dynamic Thermal Management for Internet Data Centers," *IEEE Internet Computing,* vol. 9, no. 1, pp. 42-49, Jan./Feb. 2005.

[40] D. Agonafer, L. Gan-Li, and D.B. Spalding, "LVEL Turbulence Model for Conjugate Heat Transfer at Low Reynolds Numbers," *Am. Soc. Mechanical Eng., EEP, Application of CAE/CAD to Electronic Systems,* 1996.

[41] K.K. Dhinsa, C.J. Bailey, and K.A. Pericleous, "Turbulence Modelling and Its Impact on CFD Predictions for Cooling of Electronic Components," *Proc. Ninth Intersoc. Conf. Thermal and Thermomechanical Phenomena in Electronic Systems,* 2004.

[42] M.K. Patterson, X. Wei, and Y. Joshi, "Use of Computational Fluid Dynamics in the Design and Optimization of Microchannel Heat Exchangers for Microelectronics Cooling," *Proc. ASME Summer Heat Transfer Conf.,* 2005.

[43] G. Xiong, M. Lu, C.L. Chen, B.P. Wang, and D. Kehl, "Numerical Optimization of a Power Electronics Cooling Assembly," *Proc. 16th IEEE Applied Power Electronics Conf. and Exposition,* 2001.

[44] J. Choi, Y. Kim, A. Sivasubramaniam, J. Srebric, Q. Wang, and J. Lee, "Modeling and Managing Thermal Proles of Rack-Mounted Servers with ThermoStat," *Proc. 13th Int'l Symp. High-Performance Computer Architecture,* Feb. 2007.

[45] "Intel P4 Power Measure," http://www.lostcircuits.com/, 2006.

[46] "Intel Xeon Processor," http://www.intel.com/design/xeon/, 2008.

[47] "Power Supply," http://www.energystar.gov, Summary of Rationale for Version 1.0 ENERGY STAR External Power Supply (EPS) Specification Sept. 2005, 2008.

[48] "Phoenics User Manual for Program Version 3.6.,"CHAM, http://www.cham.co.uk/, 2008.

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.

[49] "Thermal Sensor DS18B20," http://www.maxim-ic.com, 2008.

[50] "SPCR's Unique Heatsink Testing Methodology," http://www.silentpcreview.com/Sections+index-req-printpage-artid-46.html, 2002.

[51] J. Srebric and Q. Chen, "An Example of Verification, Validation, and Reporting of Indoor Environment CFD Analyses," *ASHRAE Trans.,* vol. 108, no. 2, pp. 185-194, 2002.

[52] K. Dhinsa, C. Bailey, and K. Pericleous, "Investigation into the Performance of Turbulence Models for Fluid Flow and Heat Transfer Phenomena in Electronic Applications," *IEEE Trans. Components and Packaging Technologies,* vol. 28, no. 4, pp. 686-699, Dec. 2005.

[53] Computer Systems Laboratory, Pennsylvania State Univ., http://csl.cse.psu.edu/, 2008.

[54] M.J. Crippen et al., "BladeCenter Packaging, Power, and Cooling," *IBM J. Research and Development,* vol. 49, no. 6, pp. 887-904, 2005.

[55] C.D. Patel, C.E. Bash, R. Sharma, A. Beitelmal, and C.G. Malone, "Smart Chip, System and Data Center Enabled by Advanced Flexible Cooling Resources," *Proc. 21st Ann. IEEE Semiconductor Thermal Measurement and Management Symp.,* pp. 78-85, 2005.

[56] S. Guggari, D. Agonafer, C. Belady, and L. Stahl, "A Hybrid Methodology for the Optimization of Data Center Room Layout," *Proc. Pacific Rim/ASME Int'l Electronic Packaging Technical Conf. and Exhibition,* pp. 605-612, July 2003.

Jeonghwan Choi received the BS degree in computer science, the MS degree in information and communication engineering, and the PhD degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST) in 1990, 1997, and 2007. Since 2007, he has been a postdoctoral researcher at Pennsylvania State University. His research interests include power management, system virtualization, and thermal management.

Youngjae Kim received the BS degree from Sogang University in 2001 and the MS degree from the Korea Advanced Institute of Science and Technology in 2003. From 2003 to 2004, he was a researcher at ETRI Korea. He is currently a PhD student in the Computer Science and Engineering Department at Pennsylvania State University. His research interests include operating systems and power and thermal management of storage systems.

Anand Sivasubramaniam received the BTech degree in computer science from the Indian Institute of Technology, Madras, in 1989 and the MS and PhD degrees in computer science from the Georgia Institute of Technology in 1991 and 1995, respectively. Since Fall 1995, he has been with the faculty of Pennsylvania State University, where he is currently a professor. His research interests include computer architecture, operating systems, performance evaluation, and applications for both high-performance computer systems and embedded systems. His research has been funded by the US National Science Foundation (NSF) through several grants, including the NSF Faculty Early Career Development (CAREER) Award, and from industries, including IBM, Microsoft, and Unisys Corp. He has several publications in leading journals and conferences, is on the editorial board of the *IEEE Transactions on Parallel and Distributed Systems*, and served on the editorial board of the *IEEE Transactions on Computers.* He is a recipient of the 2002, 2004, and 2005 IBM Faculty Awards. He is a member of the IEEE, the IEEE Computer Society, and the ACM.

Jelena Srebric received the BS and MS degrees from the University of Belgrade and the PhD degree from the Massachusetts Institute of Technology. She is an associate professor of architectural engineering and an adjunct professor of mechanical and nuclear engineering at Pennsylvania State University (PSU). She conducts research and teaches in the field of building energy consumption, air quality, and ventilation methods. She designed and built a state-of-the-art environmental chamber facility at PSU for energy and indoor air quality studies. Her work is sponsored by several grants form the US National Science Foundation (NSF) and the US National Institute for Occupational Safety and Health (NIOSH). She is a recipient of both NSF and NIOSH career awards. She has published extensively in the field and received several research awards, including an award from the International Academy of Indoor Air Sciences.

Qian Wang received the BS degree in mechanical engineering from Peking University, Beijing, in 1992 and the MA and PhD degrees in mechanical and aerospace engineering from Princeton University in 1997 and 2001, respectively. From 2001 to 2002, she was a postdoctoral researcher in the Storage System Department at Hewlett-Packard Laboratories, Palo Alto, California. In Fall 2002, she joined the Mechanical Engineering Department at Pennsylvania State University as an assistant professor. Her research interests include robust control, nonlinear control, and optimization, with applications to aerospace, mechanical, and computer systems. She is a member of the IEEE, ASME, AIAA, and Sigma Xi. She was a recipient of the James L. Henderson Jr. Memorial Professorship from the College of Engineering at Pennsylvania State University.

Joonwon Lee received the BS degree from Seoul National University in 1983 and the PhD degree from the Georgia Institute of Technology in 1991. After working for IBM, he joined the faculty of the Korea Advanced Institute of Science and Technology (KAIST) in 1992. His research interests include low-power computing, virtual machines, and thermal management.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 28, 2008 at 19:24 from IEEE Xplore. Restrictions apply.