Modeling and Managing Thermal Profiles of Rack-mounted Servers with *ThermoStat*

Jeonghwan Choi* Youngjae Kim Anand Sivasubramaniam
Dept. of Computer Science and Engineering,
Pennsylvania State University, University Park, PA 16802.

Jelena Srebric, Dept. of Architectural Eng., Pennsylvania State Univ., University Park, PA

Qian Wang, Dept. of Mechanical Eng., Pennsylvania State Univ., University Park, PA

Joonwon Lee, Division of Computer Science, KAIST, Daejeon, Korea

Abstract

High power densities and the implications of high operating temperatures on the failure rates of components are key driving factors of temperature-aware computing. Computer architects and system software designers need to understand the thermal consequences of their proposals, and develop techniques to lower operating temperatures to reduce both transient and permanent component failures. Until recently, tools for understanding temperature ramifications of designs have been mainly restricted to industry for studying packaging and cooling mechanisms, with little access to such toolsets for academic researchers. Developing such tools is an arduous task since it usually requires cross-cutting areas of expertise spanning architecture, systems software, thermodynamics, and cooling systems. Recognizing the need for such tools, there has been recent work on modeling temperatures of processors at the micro-architectural 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.

This paper presents a detailed 3-dimensional Computational Fluid Dynamics based thermal modeling tool, called ThermoStat, *for rack-mounted server systems. Using this tool, we model a 20 (each with dual Xeon processors) node rack-mounted server system, and validate it with over 30 temperature sensor measurements at different points in the servers/rack. We conduct several experiments with this tool to show how different load conditions affect the thermal profile, and also illustrate how this tool can help design dynamic thermal management techniques.*

*This work was conducted while Jeonghwan Choi visited PSU during his doctoral program at Korea Advanced Institute of Science and Technology (KAIST).

1. Introduction

Growing power densities are making thermal consideration a first class citizen in the design and deployment of next generation servers and datacenters. 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 alternate 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 [15, 8]. 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, that are becoming more prevalent in machine rooms and datacenters. 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 these factors are pointing to the need for designing systems for the average/common case behavior, with dynamic thermal management techniques (DTM) stepping in when thermal emergencies are encountered. Such a design philosophy requires an in-depth understanding of several inter-related cross-domain topics covering computer architecture/circuits, systems software, thermodynamics, fluid dynamics, packaging, etc. Further, it requires cross-cutting tools where one can study different interactions - workloads, temperature, air flow, system/room geometries, etc. 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 from the other. Instead, designing such adaptive systems and DTM techniques requires a closer harmony between these domains with tools that each can use

1-4244-0805-9/07/$25.00 ©2007 IEEE

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 (such as [43, 46] for processors, and [15] for disk drives) that are being used by system designers for architectural/software innovations [43, 23, 44, 16, 22, 46] 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 industry - mainly for packaging studies and rating machine ambient temperatures - with most of them being proprietary and not readily available to the academic/research community.

A recent utility [17] has been proposed to emulate 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 3-dimensional profile of the temperature within the system. We present a server and rack level thermal modeling tool called *ThermoStat* (for *Thermal-Statistics*) 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 3-D 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 break-down, etc.) occur, and how long it takes for such change. It can thus be integrated with other performance-power simulators [7, 10] used by the architecture/systems community for integrated studies, or can be run in stand-alone mode after obtaining the required values from those simulators. We have modeled 20 node rack-mounted servers using this tool, and have validated it by comparing the predictions with temperature readings from over 30 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 best to put together the underlying components, ThermoStat can be used in static settings to determine (i) where components (processors, memory, NICs, disks, etc.) need to be located within a server, where fans (and their CFMs) need to be placed, and (ii) how to place the servers, network switches and disk arrays within a rack, and designing 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 [15]), and 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" dynamic thermal management techniques as described below:

- Until now, dynamic thermal management 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 inter-related, 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 is essential for undertaking such studies, which is typically unavailable on an infrastructure which only provides temperature sensors (which is the case on an actual platform).
- Pro-active thermal management can be a better alternative than a purely reactive option in several situations. For instance, rather than wait for the temperature to reach a threshold before taking remedial actions after a temperature impacting event (e.g. fan break-down), better runtime mechanisms could be employed if we knew (i) whether the temperature will in fact reach emergency proportions, and (ii) how long it would take to reach that point. Pro-actively 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 to get there, and what is the best recourse for those conditions.
- 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 the growing energy costs, revenue based thermal management becomes extremely important for next generation datacenters [5, 9].

The rest of this paper is organized as follows. The next section points out the related work. An overview of the underlying philosophy behind ThermoStat's design is discussed in section 3. Section 4 gives details of the CFD modeling and the configurable parameters, with the validation results given in section 5. Metrics for comparing thermal profiles are discussed in section 6, and results with different configurations together with an illustration of the use of this tool for thermal management are given in section 7. Finally, section 8 concludes with directions for future work.

2. Related Work

Thermal Modeling of Specific Components: 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 [43] for microprocessors, which models temperature using thermal resistances and capacitances derived from the layout of micro-architectural 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 [4, 46]. Such estimation has been used for developing temperature aware scheduling [46]. There have also been thermal modeling studies for individual disks [15] and disk arrays [18], with the former providing a tool which 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 [6, 43, 13, 40, 14, 22, 35, 48] in the architecture community for reducing power/temperature.

All 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. A recent tool [17] 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 [17], a CFD based model is needed for a more holistic examination of the system under a wider spectrum of static (e.g. where to place components, fans?) and dynamic (e.g how long before the temperature reaches a threshold upon fan failure? what thermal management technique provides the best recourse upon emergency?). These issues are elaborated further in the paper. Fluid flows need to be modeled accurately for figuring out where components need to be placed and understanding complete system interactions.

Thermal Modeling of Datacenters: The importance of cooling high density datacenters/machine-rooms has attracted considerable interest recently [38, 37, 42, 2, 39, 24]. Most of these studies (e.g. [21, 32, 30, 29, 3, 31, 28]) have looked at this problem from an engineering perspective of designing CRAC and other cooling systems, placement of racks in machine rooms, etc., with many of them using CFD models. For instance, [30] points out that heat recirculation is a limiting factor in existing cooling systems and proposes using heat exchangers in the ceiling. Impact of CRAC failures on static provisioning has also been studied using CFD models [31]. From the computer science/systems perspective, researchers are starting to use CFD models for workload placement [25, 26, 27] across racks of a machine room, and balancing the temperature across these racks [41].

Our work is intended to provide the tools for bridging the gap between these two granularity of thermal models - those at the individual component level (within processors, disks), and those at the machine room level (comprising multiple racks) - for conducting both static and dynamic thermal management studies. Though such tools do 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. Rationale for Methodology

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

- Sensor measurements can be inaccurate in both spatial and temporal dimensions. Sensor placement to find hospots is an extremely hard problem. Further, 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 just at the points where thermal emergencies need to be monitored, but even at other spatial regions which can affect the temperature at these points (which may be needed for pro-active control). Densely filling the 3-dimensional space with temperature sensors is an infeasible and 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. Further, one may need these thermal studies to be performed at the design stage, before the physical realization.

Some of these issues - such as the last point about cost of building and conducting extensive tests on actual platforms - are not unique to thermal modeling, and simulators have traditionally been used to address such concerns. Even though a field test 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.

There could be 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 micro-architectural simulator (sim-outorder). We could even have finer resolution models going down to 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 as well as the position/parameters of power sources and fans. Such a level of modeling is usually done through Computational Fluid Dynamics (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, [4, 46] 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 was 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 8). We still believe in using a CFD-based approach for ThermoStat because of 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 take can then be stored in a database for consultation at runtime. Finally, ThermoStat can be a way for validating other temperature measurement (using sensors) or modeling (as in [17, 4, 46]) techniques, and can be used in conjunction with those to develop hybrid multi-resolution models.

4. CFD Modeling

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

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

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 x, y or z direction; Γ is the diffusion coefficient; 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 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.

Identifying a suitable turbulence model is very important for the accuracy of CFD simulations. ThermoStat uses the LVEL model [1], 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 low Reynolds number flow regime, and, therefore, k-ϵ model is not a suitable choice. A study [12] tested seven different turbulence models including the standard k-ϵ model and LVEL to find that the tested models performed better than k-ϵ model, and that LVEL even though is the simplest one, was as effective as the much more complicated turbulence models. This finding is very useful because significant computation time (factor of three or higher based on the software packages and simulation setting) 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.

While researchers and students with backgrounds in mechanical engineering, thermodynamics and fluid mechanics, are well-versed with 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 amongst computer scientists/engineers, by hiding as many non-essential 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.), 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 speeds of fans may change, 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 boxes, locational information of CPUs/fans/disks/power-supplies etc., their operating power characteristics, inlet air temperature, etc. Further, 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. Further, we can also have default configuration files for the rack(s) that we have modeled. We believe 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. [33, 47]) 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.

Rack Parameters	
Physical Dimension (cm^3)	66 x 108 x 203 (42U)
Grid Cells (#)	45 x 75 x 188
Velocity & Pressure	On
Energy Equation	Temperature Total
Turbulence Model	LVEL
Domain Material	Ideal Gas Law
Gravitational Force	On
Buoyancy Model	Boussinesq
Coeff. for Auto Wall Func.	Log-law
Iterations (#)	5000

Component	Size (cm)			Power (W)		Slot number
	X	Y	Z	Min	Max	(from bottom)
X335 × 20	44	66	4	110	350	4-20, 26-28
X345 × 2	44	70	9	100	660	24-25,36-37
Exp300 (14 Disks)	44	52	13	280	560	38-40
Cisco Catalyst4000	44	30	27	-	530	29-34
Myrinet(M3-32P)	44	44	13	-	246	1-3

x335 Server Box 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
Coeff. for Auto Wall Func.	Log-law
Iterations (#)	3500
Outlets (#)	3

Component	Property	Value
CPU [19] × 2	Material	Copper
	Heat Src.	31-74 (W)
Disk	Material	Aluminium
	Heat Src.	7-28.8 (W)
Power Supply [36]	Material	Aluminium
	Heat Src.	21-66 (W)
NIC	Material	Copper
	Heat Src.	2 x 2 (W)
Fans × 8	Type	Circular
	Flow Rate	0.001852 - 0.00231 (m^3/sec)

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

Table 1. Simulation Parameters.

In this paper, we have modeled and present results for a 42U rack with the layout of the slots in this rack given in Table 1. Currently, we have only modeled the twenty IBM x335 servers on this rack, and modeling of the storage array, network (Myrinet and Cisco Gigabit Ethernet) switches, and the two management nodes (x345) is part of future work. Each x335 server (Table 1) has dual 2.8GHz Xeon processors, each with a maximum power rating of 84 W when executing. 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 [20]). 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 1 (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. Further, 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 a SCSI disk, Myrinet NIC, eight fans, and a power supply, whose layout is given in Figure 1, and the associated modeling parameters are given in Table 1. The eight circular fans direct most of the air flow in the box, taking in the air through vents in 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 rack are not being modeled for simplicity, and we found that these do not significantly impact temperature within each server box. The number of grid cells and iteration counts for running the simulations have been set after experimentally determining trade-offs between speed and accuracy.

Most academic institutions have licenses for popularly used CFD software such as FLUENT, FLOTHERM, Phoenics, etc. We are currently using Phoenics [34] (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.

5. Validation

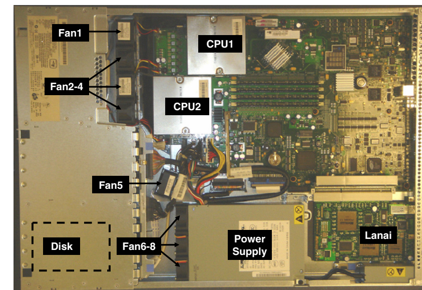
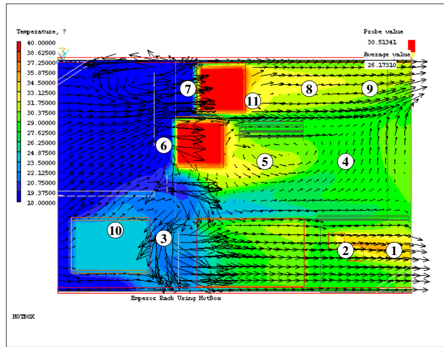


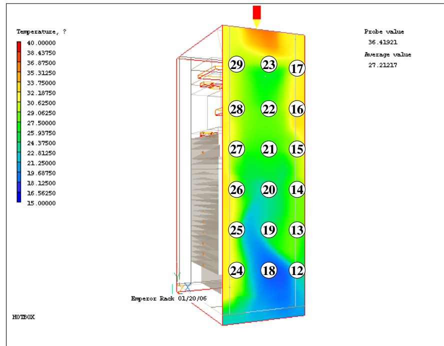
Figure 1. IBM x335 Server

To validate our CFD model, we deployed several temperature sensors (DS18B20 from Dallas Semiconductor [45]) at different points in our rack-mounted system - both within the individual x335 server boxes and at the rear (inside) of the rack whose temperatures are affected by the individual server boxes - and compared those readings with the predicted temperatures by our CFD model at those points. Figures 2 (a) and (b) show the placement of 29 sensors within the server box and at the rear of the rack. 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 or from the rear door of the rack. Two of the sensors - 10 and 11 - were stuck to the surfaces of the disk and CPU1 respectively with 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 and thus the temperatures there are expected to be lower than at the center of the CPU surface. We are currently trying to obtain surface temperature at the center using very thin thermocouples.



(a) Within a x335 Server Box



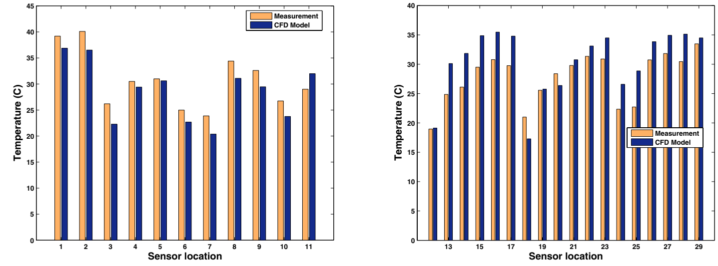
(b) Back (Inside) of Rack

Figure 2. Validation: Sensor Placement Locations within a server box and at the back of rack. Note that the color coding for the temperatures is for a cross-section (mainly air) of the shown spatial extent, and do not necessarily reflect surface temperatures of components.

We are not showing all the validation results with different operating conditions and power consumption values, in the interest of space, and they will be referred to a technical report in the final version. Instead, here we show the validation results when 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 Figure 3.

When we examine the results within the server box, 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%. 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 $\pm$ 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 is still bound to be some errors/distortions in the spatial locations of where we are measuring the data.



(a) Within server box

(b) Back of rack

Figure 3. Validation: Comparing temperature from CFD Modeling and Sensor Measurements.

When we move from within the box to the back of rack results (Figure 3 (b)), we notice the errors are slightly higher (11.00% on average). Mostly, the results from CFD across the locations of a rack are slightly higher than actual measurements except for a few points (such as sensors 18 and 20). This is because we have currently only modeled the x335 servers, and not modeled the terminal servers, network switches, and Disk array which are also present on our rack which constitute higher measurements at those locations. In addition to these sensor measurements, we also took a thermal image using an infrared camera of the back of the x335 cases (surface temperature), and we found that the thermal profiles are quite close to that predicted by the CFD model (not shown here due to space limitations).

6. Metrics for Thermal Profile Comparison

One of the issues in thermal studies is figuring out how to compare between two thermal profiles for the same space under consideration (say when we want to find out how an architectural change impacts the system). A CFD model gives the temperature at each point in the 3-dimensional space, and we need ways of comparing them across two different executions:

- *Specific Points:* One option is to focus on specific points (a single point on the CPU, disk, network card, etc.) and compare the temperatures at these points in the two profiles. This is a reasonable option when the study is focused on specific components, and if one is aware of the specific points on these components that are most important to consider/study (i.e. reliability is most influenced by temperature at those points). However, one may sometimes be interested in lowering the

ambient operating temperature and it is not clear that results for specific points can paint an accurate picture.

- *Mean and Standard Deviation:* We could also consider aggregate metrics such as mean and standard deviation of the temperature across the entire 3-dimensional space. Though this information can give aggregate behavior, it can fall short in gleaning information about specific spatial regions.

- *Cumulative Spatial Distribution Function:* Rather than a single aggregate metric, one could consider the distribution of temperatures in space as a CDF, i.e. percentage of the spatial extent (on the y-axis) which is less than a certain temperature (on the x-axis).

- *Spatial Difference:* We can also consider the temperature difference at each grid point of the spatial extent between the two thermal profiles.

Case	Inlet (C)	CPU1 (GHz)	CPU2 (GHz)	Disk	Fans
1	32	1.4	1.4	Max	Fans 1-8 (Low)
2	32	2.8	Idle	Max	Fans 1-8 (High)
3	18	2.8	2.8	Max	Fan 1 (Fail), Fans 2-8 (High)
4	18	2.8	2.8	Idle	Fan 1-8 (Low)

Table 2. Synthetically Created Conditions

Case	CPU1	CPU2	Disk	Average	Std. Dev.
1	57.16	57.20	53.74	44.0	7.5
2	75.42	50.05	49.86	42.6	8.9
3	73.34	61.93	36.63	33.8	13.9
4	66.16	65.07	24.38	33.9	13.0

Table 3. Metrics (in C) for comparing the Conditions

To illustrate these metrics, we study the thermal profiles of four different system configurations/cases (see Table 2) where we consider different possibilities for CPU operation (frequencies of 2.8 and 1.4 GHz, and idle state), the disk operating at full power (28.8 W) and being completely idle (7 W), two different inlet air temperatures (the manufacturer suggests operating up to 32 C), and two different speeds for the fans (with Fan 1 failing in one of the configurations). In the case of the CPU power, at the lower frequency we assume a simple frequency scaling model without any voltage scaling, i.e. the power is linearly proportional to the frequency, and use the maximum thermal design power (74 W that is assumed at 2.8 GHz [19]) to calculate the power for lower frequencies. The shown CPU temperatures are for the center of the CPU surface.

As we see from Table 3, the temperature of individual components (at specific points) is largely dependent on the power consumption of that component. However, we note that the temperature is also influenced by external issues such as the inlet temperature. For instance, note that the temperature of CPU1 went from 66 C (Case 4) to 75 C (Case 2) when the inlet air temperature went from 18 C to 32 C, despite the fans going faster. The breakdown of a Fan 1 also causes a sharp rise in CPU1 (which is closest to this fan) temperature. These observations motivate the need for studying complete system interactions, over and beyond thermal studies of individual components in isolation.

While the temperatures at specific points are useful to study the influence of different operating conditions on those components, one needs to know what to look for (in terms of grid points) when making such observations. Looking at just the average and standard deviation of the temperatures across the grid points may not necessarily give much insights. For instance, between Case 3 and Case 4, the changes in fan operation hardly change the average and standard deviation, while they do change the CPU1 temperature considerably. On the other hand, a change in the inlet temperature substantially affects even the aggregate values. These observations are also evident in the Cumulative Spatial Distribution Function graph in Figure 4 (a), where the curves for Case 1 and Case 2, are pushed more to the right than for the other two cases. Still this CDF graph gives more information than just an average - even though the averages for Cases 3 and 4 are comparable, the CDF graph for Case 3 is more to the right, indicating more regions of higher temperature.

Finally, Figures 4 (b) and (c) show the spatial pairwise difference graphs, which are much more revealing than the above aggregate metrics. In Figure 4 (b), we see that the higher fan speeds and idle CPU2 cause the temperature to go down across most of the box, except close to CPU1 whose power consumption has increased. Similarly, Figure 4 (c) shows the higher temperature in the region affected by the failure of Fan 1 in Case 3 compared to Case 4. These difference graphs are very useful to understand how thermal profiles change, together with pinpointing the spatial regions affected by such change.

7. Using ThermoStat

There are several ways of using ThermoStat for static machine design as well as developing dynamic thermal management techniques. Below we give some illustrative examples.

7.1. Are servers in a rack independent?

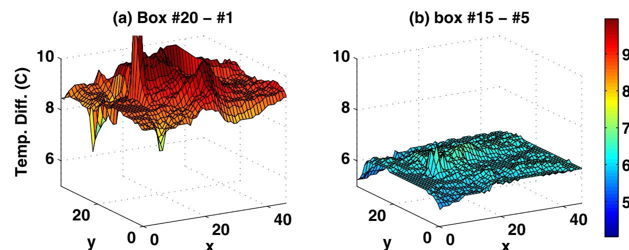
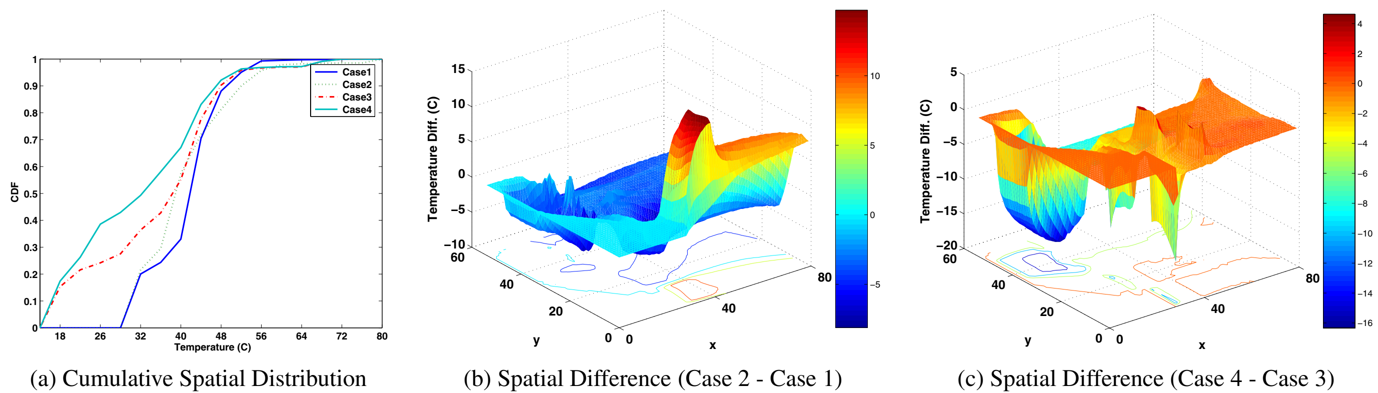


Figure 5. Temp. Diff. between servers of a Rack

It is interesting to see how machines in a rack, if at all, do influence each other's temperature. 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



(a) Cumulative Spatial Distribution (b) Spatial Difference (Case 2 - Case 1) (c) Spatial Difference (Case 4 - Case 3)

Figure 4. Comparing the Metrics for thermal profiles for the considered cases

front, and as is to be expected, it is hotter nearer the top. We picked four machines - 1, 5, 15, 20 (increasing order from bottom of rack) - for comparing the thermal profiles. All these machines are in the idle mode. Figure 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 Figure 5 (b), 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. assign higher load to machines at the bottom of the rack.

7.2. Are components in a server independent?

Static design considerations when packaging components within a server box include understanding (i) the range of inlet temperatures for safe operation of components, (ii) whether the provisioned fans are able to adequately cool the components, and (iii) how the heat generated by components interact with each other (i.e. are they laid out properly). In the previous section, we already studied issues related to inlet temperature and fan operation, and in Figure 6 we examine how components, if at all, affect each other's temperatures. 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 operating at maximum power. In addition to temperatures of individual components, the graph also plots the average temperature within the server box. As the results in this graph show, even though the average temperature of the spatial extent does change with the load on the components, components exhibit little interaction between each other on the modeled system. This is because of the design of the x335, where the components are laid out fairly well apart (see Figure 1), and 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 already showed in the previous section that the component temperatures are significantly impacted by the fans - the air flow directed by them - and one should not misunderstand the results in Figure 6 to imply that each component's temperature is dependent only on its own characteristics (power, materials, etc.).

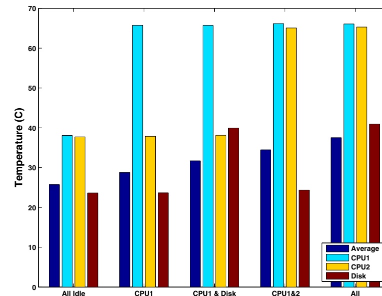


Figure 6. Examining interactions, if any, between components. Legends on x-axis indicate which components are active (running at maximum power) with rest being idle.

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 cost. We are already seeing sophisticated layouts and airflow techniques in dense blade servers. For instance, in IBM's HS20 blade server [11], the two CPUs occupy nearly a third of the floor area, making it very difficult to avoid the air flowing from one to the other. The air inlet is not in the front for this system, and is near a memory bank instead. Further, 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 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 micro-architectural management of temperature is becoming important, over and beyond packaging optimizations.

7.3. Designing DTM Techniques

ThermoStat can also be used for designing and evaluating Dynamic Thermal Management (DTM) techniques. We illustrate this below with two examples to show how ThermoStat can help design both reactive and pro-active DTM techniques for controlling CPU temperature.

7.3.1 What should we do when a fan breaks? - A Reactive Example

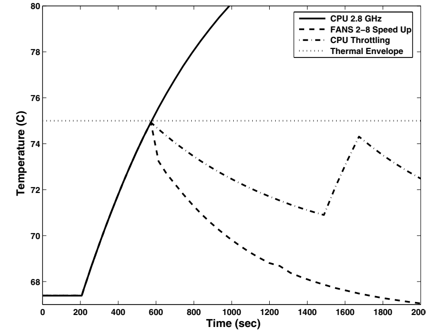
In this example, we make Fan 1 breakdown at time 200 seconds (see Figure 7 (a)), causing the CPU1 temperature to start rising rapidly. The thermal envelope of safe CPU operation is set to 75 C (from [19]), and if there is no management technique, ThermoStat shows us that the CPU temperature running at 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 - 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 to take for controlling its temperature. In Figure 7 (a), we consider two possible reactive measures when reaching this threshold. The first option is to make all 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/sec, and we change this to 0.00231 m^3/sec. As we can see, this does compensate for any rise in temperature, which is again information that would not be available without modeling air flow. The other reactive measure we consider is cutting down the CPUs operating frequency by 25%, i.e. it now runs at 2.1 GHz, which is also effective at 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 1500 secs), and so on. Between these two options, the former may be preferable if performance is more critical since this option does not loose any CPU capacity.

In this example, ThermoStat helps us identify the possible reactive options, evaluate their effectiveness, and quantify times for getting to these associated temperatures.

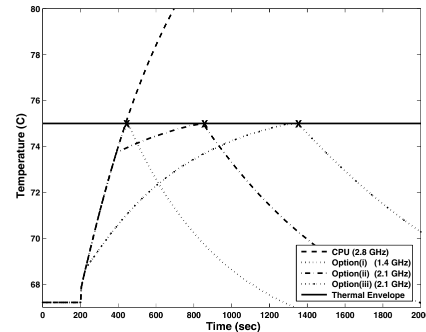
7.3.2 What should we do when inlet air temperature suddenly rises? - A Pro-active Example

In this example, we make the inlet air temperature suddenly go up to 40 C from 18 C at time 200 secs as shown in Figure 7 (b). Though such instantaneous change is somewhat drastic - machine room temperatures do 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 wait until reaching the thermal envelope, at which point we may have waited too long, one may want a more pro-active thermal management strategy, i.e. take remedial actions before the emergency point. At the same time, taking the actions too early - say immediately after



(a) Fan 1 fails at time 200 secs.



(b) Inlet Air temp. suddenly changes from 18 C to 40 C at time 200 secs.

Figure 7. Designing DTM Techniques with ThermoStat

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 C operating conditions, scaling back the CPU frequency by 25% does not really keep the temperature within the envelope, and we use a 50% 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: (i) Running the CPU at full frequency until the emergency point, at which point (time = 440 secs) scaling back the frequency by 50% which is what the purely reactive approach would do; (ii) Running the CPU at full frequency for another 190 seconds after detecting inlet air temperature change, then (at time = 390 secs) resorting to 25% frequency scale back, and then later when reaching emergency (at time = 821 secs) cutting the CPU frequency further to 50% of maximum value; and (iii) Running the CPU at full frequency for another 28 seconds, then (at time = 228 seconds) scaling back the CPU by 25%, and then scaling back the CPU to 50% when reaching emergency threshold at time 1317 secs. The choice of which option to use depends on the workload. For instance, if the amount of work remaining to be done requires 500 secs when operating at full speed, the three options would complete this job at times 960, 803 and 857 seconds respectively, making option (ii) 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.

8. Concluding Remarks

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 to place components, design cooling systems, etc. - as well as for undertaking higher level (architectural/software) thermal optimization studies. Until now, such tools have been mainly restricted to industry, and ThermoStat is intended to fill this void in the research/academic community. We anticipate to release 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.

Even though running ThermoStat in real-time with system execution may not be an option (which can be one of the advantages of [17]) because of the high simulation time, we envision that it will serve a similar role as architectural tools that have been extensively used for systems design over the years, for evaluating different alternatives ("what-if" questions). The simulation cost for such studies with ThermoStat is comparable to those experienced with popularly used architectural simulators. For instance, with the grid parameters and iteration counts given in section 4, obtaining a temperature profile for a single server box takes roughly 20-30 minutes on an AMD Athlon64 machine with 1 GB memory. If we assume the granularity of simulated time for a data-point to be 20-30 seconds (as we see in Figure 7, temperatures take several seconds to change for the envisioned system events that we may need to handle), we see that ThermoStat has between 40X to 90X slowdown for single box studies. Studying a complete rack for a similar time granularity incurs around 400-500X slowdown. However, faster machines and/or employment of parallelism (there are several parallel CFD platforms which we could use), or coarser time resolutions for simulation (based on time it takes for temperatures to change and the granularity of control), could considerably lower this overhead. Further, even if there are some absolute differences between machines of a rack based on position, the relative trends within a machine are similar. Consequently, we may be able to start with slightly adjusted boundary conditions to mimic the behavior of a machine in the rack, while still performing the simulations of a single machine. We are investigating these issues currently.

In addition, we also 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, inlet temperatures) is not expected to be excessively high, and we are currently trying to build such a database of events for our system together with narrowing down the information that would be needed for runtime decision making. Such an approach can help us integrate this framework with other sensor-based/runtime temperature management mechanisms for multi-resolution temperature measurement/management. We are also examining interfaces between this framework and other architectural performance/power/thermal modeling tools being used by the community. Finally, we are looking to incorporate the network switches and disk array into our rack model, together with developing and validating models for denser blade-based racks.

Acknowledgements

This research is supported in part by NSF grants 0429500, 0325056, 0509234 and 0615097.

References

[1] D. Agonafer, L. Gan-Li, and D. B. Spalding. LVEL turbulence model for conjugate heat transfer at low Reynolds numbers. *American Society of Mechanical Engineers, EEP, Application of CAE/CAD to Electronic Systems*, 1996.

[2] C. E. Bash, C. D. Patel, and P. K. Sharma. Efficient Thermal Management of Data Center - Immediate and Long-Term Research Needs. *Intl. J. Heat, Ventilating, Air-Conditioning and Refrigeration Research Needs*, 9(2):137–152, 2003.

[3] M. H. Beitelmal and C. D. Patel. Thermo-Fluids Provisioning of a High Performance High Density Data Center. *HPL 2004-146 (R.1), HP Lab. Technical Report*, 2004.

[4] F. Bellosa, S. Kellner, M. Waitz, and A. Weissel. Event-driven energy accounting of dynamic thermal management. In *Proceedings of the Workshop on Compilers and Operating Systems for Low Power*, September 2003.

[5] R. Bianchini and R. Rajamony. Power and Energy Management for Server Systems. *IEEE Computer*, 37(11), November 2004.

[6] D. Brooks and M. Martonosi. Dynamic Thermal Management for High-Performance Microprocessors. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, pages 171–182, January 2001.

[7] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A Framework for Architectural-Level Power Analysis and Optimizations. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 83–94, June 2000.

[8] S. Charrap, P. Lu, and Y. He. Thermal Stability of Recorded Information at High Densities. *IEEE Transactions on Magnetics*, 33(1):978–983, January 1997.

[9] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle. Managing Energy and Server Resources in Hosting Centers. In *Proceedings of the 18th ACM Symposium on Operating System Principles (SOSP)*, October 2001.

[10] W. Chen, M. Dubois, and P. Stenstrom. Integrating Complete-System and User-level Performance/Power Simulators: The SimWattch Approach. In *Proceedings of the International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2003.

[11] M. J. Crippen et al. BladeCenter packaging, power, and cooling. *IBM J. Res. & Dev.*, 49(6):887–904, 2005.

[12] K. K. Dhinsa, C. J. Bailey, and K. A. Pericleous. Turbulence modelling and it's impact on CFD predictions for cooling of electronic components. In *Thermomechanical Phenomena in Electronic Systems -Proceedings of the Ninth Intersociety Conference*, 2004.

[13] M. Gomaa, M. D. Powel, and T. N. Vijaykumar. Heat-and-Run: Leveraging SMT and CMP to Manage Power Density

Through the Operating System. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 260–270, 2004.

[14] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPM: Dynamic Speed Control for Power Management in Server Class Disks. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 169–179, June 2003.

[15] S. Gurumurthi, A. Sivasubramaniam, and V. Natarajan. Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 38–49, June 2005.

[16] J. Hasan, A. Jalote, T. N. Vijaykumar, and C. Brodle. Heat Stroke: Power-Density-Based Denial of Service in SMT. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, pages 166–177, 2005.

[17] T. Heath, A. P. Centeno, P. George, Y. Jaluria, and R. Bianchini. Mercury and Freon: Temperature Emulation and Management in Server Systems. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, October 2006.

[18] R. Huang and D. Chung. Thermal Design of a Disk-Array System. In *Proceedings of the InterSociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pages 106–112, May 2002.

[19] Intel Xeon Processor. http://www.intel.com/design/xeon/.

[20] Intel P4 Power Measure. http://www.losticircuits.com/.

[21] J. F. Karlsson and B. Moshfegh. Investigation of indoor climate and power usage in a data center. *Energy and Buildings*, 37:1075–1083, 2005.

[22] Y. Kim, S. Gurumurthi, and A. Sivasubramaniam. Understanding the Performance-Temperature Interactions in Disk I/O of Server Workloads. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, pages 179–189, Febuary 2006.

[23] Y. Li, K. Skadron, Z. Hu, and D. Brooks. Performance, Energy, and Thermal Considerations for SMT and CMP Architectures . In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, pages 71–82, Febuary 2005.

[24] J. Moore, J. Chase, , and P. Ranganathan. ConSil: Low-cost Thermal Mapping of Data Centers. In *Proceedings of the Workshop on Tackling Computer Systems Problems with Machine Learning Techniques (SysML)*, June 2006.

[25] J. Moore, J. Chase, , and P. Ranganathan. Weatherman: Automated, Online, and Predictive Thermal Mapping and Management for Data Centers. In *Proceedings of the International Conference on Autonomic Computing (ICAC)*, June 2006.

[26] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making Scheduling Cool: Temperature-Aware Workload Placement in Data Centers. In *Proceedings of the USENIX Annual Technical Conference*, April 2005.

[27] 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. In *Proceedings of the Workshop of Temperature-Aware Computer Systems (TACS-1) held at ISCA*, May 2002.

[28] S. V. Patankar and K. C. Karki. Distribution of Cooling Airflow in a Raised-Floor Data Center. In *American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE)*, 2004.

[29] C. D. Patel, C. E. Bash, and M. Beitelmal. Smart Cooling of Data Centers. In *Proceedings of the Pacific RIM/ASME International Electronics Packaging Technical Conference and Exhibition (IPACK)*, July 2003.

[30] 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 . In *Proceedings of ASME International Electronic Packaging Technical Conference and Exhibition (IPACK)* , July 2001.

[31] C. D. Patel and A. J. Shah. Cost Model for Planning, Developement and Operation of a Data Center. *HPL 2005-107 (R.1), HP Lab. Technical Report*, 2005.

[32] C. D. Patel, R. Sharma, C. E. Bash, and A. Beitelmal. Thermal Considerations in Cooling Large Scale High Compute Density Data Centers. In *Proceedings of Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, May 2002.

[33] 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. In *Proceedings of the ASME Summer Heat Transfer Conference*, 2005.

[34] *Phoenics User Manual for Program Version 3.6., CHAM Ltd.* http://www.cham.co.uk/.

[35] E. Pinheiro and R. Bianchini. Energy Conservation Techniques for Disk Array-Based Servers. In *Proceedings of the International Conference on Supercomputing (ICS)*, June 2004.

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

[37] N. Rasmussen. Cooling Strategies for Ultra-High Density Racks and Blade Servers. In *APC White Paper 46 http://www.apcc.com/prod_docs/results.cfm?class=wp &allpapers=1*.

[38] N. Rasmussen. Guidelines for Specification of Data Center Power Density. In *APC White Paper 120 http://www.apcc.com/prod_docs/results.cfm?class=wp &allpapers=1*.

[39] P. Rodgers and V. Eveloy. Prediction of Microelectronics Thermal Behavior in Electronic Equipment: Status, Challenges and Future Requirements. In *Proceedings of the International Conference on Thermal and Mechanical Simulation and Experiments in Micro-Electronics and Micro-Systems*, 2003.

[40] L. Shang, L.-S. Peh, A. Kumar, and N. Jha. Thermal Modeling, Characterization and Management of On-Chip Networks. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*, pages 67–78, December 2004.

[41] R. Sharma, C. Bash, C. Patel, R. Friedrich, and J. Chase. Balance of Power: Dynamic Thermal Management for Internet Data Centers. *IEEE Internet Computing*, 9(1):42–49, January 2005.

[42] R. K. Sharma, C. E. Bash, and C. D. Patel. Dimensionless parameters for evaluation of thermal design and performance of large-scale data centers. In *8th ASME/AIAA Joint Thermophysics and Heat Transfer Conf.*, June 2002.

[43] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-Aware Microarchitecture. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 1–13, June 2003.

[44] J. Srinivasan and S. Adve. Predictive Dynamic Thermal Management for Multimedia Applications. In *Proceedings of the International Conference on Supercomputing (ICS)*, pages 109–120, June 2003.

[45] Thermal Sensor DS18B20. http://www.maxim-ic.com.

[46] A. Weissel and F. Bellosa. Dynamic Thermal Management for Distributed Systems. In *Proceedings of the Workshop on Temperature-Aware Computer Systems (TACS)* , June 2004.

[47] G. Xiong, M. Lu, C. L. Chen, B. P. Wang, and D. Kehl. Numerical optimization of a power electronics cooling assembly. In *IEEE Applied Power Electronics Conference and Exposition*, 2001.

[48] Q. Zhu, F. David, C. Devraj, Z. Li, Y. Zhou, and P. Cao. Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, 2004.