ConSil: Low-cost Thermal Mapping of Data Centers

Justin Moore†, Jeffrey S. Chase†, and Parthasarathy Ranganathan‡

†*Duke University*
{*justin, chase*}*@cs.duke.edu*

‡*Hewlett Packard Labs*
partha.ranganathan@hp.com

Abstract

Several projects involving high-level thermal management — such as eliminating "hot spots" or reducing cooling costs through intelligent workload placement — require ambient air temperature readings at a fine granularity. Unfortunately, current thermal instrumentation methods involve installing a set of expensive hardware sensors. Modern motherboards include multiple on-board sensors, but the values reported by these sensors are dominated by the thermal effects of the server's workload.

We propose using machine learning methods to model the effects of server workload on on-board sensors. Our models combine on-board sensor readings with workload instrumentation and "mask out" the thermal effects due to workload, leaving us with the ambient air temperature at that server's inlet. We present a formal problem statement, outline the properties of our model, describe the machine learning approach we use to construct our models, and present *ConSil*, a prototype implementation.

1 Introduction

As the number of servers and power requirements of servers increase, data center designers and managers must account for factors beyond standard performance issues. Yet instrumentation of these factors that is available to management agents lags far behind that for performance and IT considerations.

The first prerequisite of integrating power and thermal concerns into a management agent is accurate and complete information to drive the management policy. A crucial component is a detailed *thermal map* of the data center, containing temperature and airflow information at a fine-grained resolution. For example, recent work in data center thermal management reveals that maintaining a low inlet temperature leads to lower cooling costs [12, 7]. Implementing this policy requires an accurate reading of the inlet temperature at every server.

Yet the amount and type of thermal data we can monitor is less fine-grained than that available for application and system performance. A management agent attempting to optimize system performance can utilize raw data from processors, memory subsystems, network devices, and storage devices, as well as application-level metadata from batch queues, web servers, and other data center applications. A management agent attempting to control thermal conditions must draw from sparse or ineffective sensors. Useful data, such as ambient air temperatures, are generally collected using a separate network of temperature sensors. These sensors, placed on rack enclosures and A/C units [9, 5], can be expensive to obtain, time-consuming to deploy, and difficult to read; it is not uncommon to have only two or three such sensors placed on a standard rack enclosure.

Other temperature sensors we can monitor, such as those on most modern motherboards, report the temperature at selected points within a server. However, these sensors do not provide a good proxy for ambient temperature since their values are influenced heavily by local thermal conditions, such as heat from the processor(s). While some servers contain a temperature sensor near the front inlet, a data center owner should not be forced to limit their purchasing options based on this single factor.

We propose constructing a thermal map that includes per-server inlet temperatures by modeling and then "masking out" local thermal conditions in each server. While a single sensor — such as those on top of each processor — may be dominated by the local thermal effects of that server's workload, the readings from multiple sensors over time allow us to model the effects of a given workload on those sensors. We leverage machine learning techniques to combine existing workload data with multiple internal temperature readings to infer the current server inlet temperature. We demonstrate the effectiveness of this approach by building thermal models for an existing line of servers. With a few hundred data points per server, our models are capable of infer inlet temperatures within 1°C over 80% of the time, and within 2°C over 98% of the time. This degree of accuracy is similar to that of off-the-shelf temperature sensors.

2 Motivation

Current-generation 1U servers consume over 350 Watts at peak utilization, releasing much of this energy as heat; a standard 42U rack of such servers consumes over 8 kW. As data centers migrate to bladed servers over the next few years, these numbers could potentially increase to 55 kW per rack [8].

A thermal management policy that considers facilities components – such as A/C units and the layout of the data center – and temperature-aware IT components can decrease cooling costs [9], increase hardware reliability [2], and decrease response times to transients and emergencies [6]. Significant recent work in data center management focuses on formulating effective thermal management policies; Multiple projects reduce data center cooling costs, such as optimizing cooling [9], minimizing global power consumptions [10, 4], and efficient heat distribution [7, 3]. These projects depend on an underlying instrumentation layer to provide the thermal map. In the absence of fine-grained thermal instrumentation, these policies must rely on simplistic heuristics, such as minimizing server power consumption, A/C return temperature, or generating a uniform exhaust profile.

We are not aware of any other work looking at software models for thermal mapping of data centers. Related work has primarily used ad-hoc collections of external sensors to monitor server inlet temperature at a few selected locations [12]. As discussed earlier, in addition to the large costs with wiring and maintenance, these approaches also suffer from inaccuracies from inadequate coverage. In lieu of actual deployments, other studies have used simulation to determine thermal maps [9, 11]. However, these simulations use complex fluid dynamics, each taking several hours of run time.

Common thermal management practices involve placing two or three sensors on the front and back of each rack. This results in less than 150 sensors providing data for up to 1000 servers. Furthermore, the total cost of deploying these or additional sensors can be prohibitive, up to $100 per sensor.

Modern motherboards, on the other hand, provide the status of multiple relevant on-board components, including fans and internal temperature sensors. These sensors improve coverage, but present other challenges; namely, that the readings provided by these sensors are heavily influenced by local heat sources such as processors and disks. A temperature sensor on a 3 GHz Pentium IV processor — using over 115 Watts at maximum utilization — can register temperatures in excess of 25°C above that of the air coming into the server.

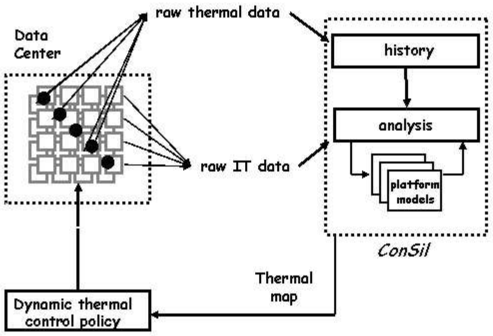

Figure 1: ConSil combines readings from internal sensors in each server platform with other instrumentation data to produce detailed thermal maps.

Symbol	Meaning
Q	Measure of heat (Watts)
W	Set of workload metrics
X	Number of workload metrics
M	Set of motherboard sensor readings
Y	Number of motherboard sensors
D	Complete set of metrics (W and M)
D_k	k^{th} most recent data set
Z	Number of recent data sets

Table 1: Parameters for problem formalization.

3 Formalizing the Problem

Figure 1 depicts how ConSil fits in to a modern data center operations infrastructure. The role of ConSil is to analyze data from internal and external thermal sensors and produce an accurate map of current thermal conditions in the data center, for input to the control policy. To extract this information, ConSil builds and applies models of heat flow in the data center and the servers it contains. It uses these models to infer the thermal map from the internal sensors in each server platform.

3.1 Problem Statement

The heat measured within our server as being the combination of the heat at the inlet of the server and the heat generated by the server's workload. Table 1 outlines the terminology and definitions we use.

$$Q_{measured} = f(Q_{inlet}, Q_{workload})$$

However, this equation omits several details. For example, most servers have multiple internal sensors. The amount of heat generated by the workload and measured by these sensors varies significantly within the server. For example, the values reported by a sensor near a processor are influenced heavily by the recent activity of that processor.

Given that workload and airflow are dynamic, it is difficult to infer the thermal effects of workload on each sensor individually. Instead, we leverage the fact that the heat measured at each sensor is the combination of heat generated by the workload and the heat from the ambient air at the server's inlet. By inferring and subtracting the common element – ambient air temperature – from measured values, we reduce the number of outputs from X to one.

While processor utilization may be the primary contributor to heat production by a server, it is not the only one. Thermodynamics tells us that all components that consume non-trivial amounts of power – including RAM, storage devices, and graphics cards – convert some of this power into heat. Our model must be able to account for these sources. In order to leverage the relationship between system utilization metrics – processor usage, memory access rates, disk read/write throughput, etc – we update our model to include workload information as a proxy for the amount of heat injected into the system.

Finally, we address the time-dependence of heat flow. While server utilization can change instantaneously, it will take time for the temperature distribution to adjust. For example, a server that has been 100% utilized will heat up; however, when that server goes idle it will take seconds or minutes for the server to eject the excess heat. Given that the current workload is constant (idle) but the internal temperature varies during this period, depending solely on current workload readings as a proxy for local thermal conditions would be unreliable. We must include recent data in order to make accurate inferences of a workload's effect on internal measurements. If we include the Z most recent data sets at time t, we can provide a formal description of our model.

$$Q_{inlet} = f\big(D_t, D_{t-1}, \ldots, D_{t-Z}\big)$$

4 ConSil

At a high level, we are dealing with a model that has $Z \cdot (X + Y)$ inputs — our workload and instrumentation data for each epoch — and one output — the inferred ambient air temperature at the server inlet.

The first step in implementing ConSil is to collect the data necessary to construct our model. Since the model is constructed off-line, it is not necessary to aggregate the data as readings are taken; it is sufficient to timestamp the reading as it is taken for later correlation. Our input data is available through a variety of standard monitoring infrastructures.

The output data — sensors that measure ambient air temperature outside the front inlets of servers — can be collected through any number of available hardware and software infrastructures. While complete coverage of the data center using these sensors alone is cost-prohibitive and complex, our method does not suffer from this limitation; we require only 10 or 15 sensors per type of server.

4.1 Machine Learning

The method we select to model heat flow and infer ambient air temperature must produce an output that falls within a continuous range of values, represent complex relationships, construct the model using a large input set, and make "live" inferences using the most recent instrumentation data. Approximate solutions that run on the order of 1 second are superior to more accurate solutions that take minutes. This class of problem benefits from the application of machine learning techniques. However, machine learning covers a broad class of methods, not all of which meet the criteria we set forth. Our criteria rule out techniques such as decision trees, tree induction algorithms, and propositional learning systems.

Neural nets, on the other hand, meet our criteria and present a reasonable analogy to the scenario at hand. Just as the strength of the relationship between particular input and output values of a neural net depends on the internal structure of the net, the correlation between workload and observed temperature depends on the physical flow of heat from server components to internal sensors. The strength of this approach is that it allows us to add observations to our model during normal operation of our servers.

For a model using the Z most recent epochs, with X internal temperature sensors and Y metrics used to characterize current system workload, there will be $Z \cdot (X + Y)$ inputs to our system. The output of our model is the inferred ambient air temperature; from there we can deduce the amount of additional heat present within the server.

It is important to note that we are not claiming neural nets are the best modeling method. Instead we show that, as an instance of a machine-learning-based approach, they have the properties we desire.

4.2 Implementation

There are several off-the-shelf neural net development libraries, enabling us to leverage these techniques rapidly. We selected the Fast Artificial Neural Net (FANN) development library [1]. FANN implements standard neural net training and execution functions, allowing us to focus on exploring effective methods of constructing our models.

We selected the sigmoid function as our neuron activation function. Next, we must determine an appropriate exponent, which controls the shape of the output distribution. A "steep" sigmoid function requires precise inputs at all layers to produce accurate outputs; small errors grow as they pass through the network. A "flat" sigmoid function may result in an overly-trained network. In other words, it can make accurate inferences for inputs similar to previously-seen data, but is not general enough to provide accurate answers for new input sets.

Before constructing our model we process our input and output data. Given that output values from the sigmoid function will be in the range $[0, 1]$ we scale all input and output values to fall within this range. This provides consistency between input and output data. We evaluate the accuracy of the models using five-fold cross-validation (FFCV) and measing the mean squared error (MSE) over the test data.

5 Results

For each type of server we collect data from external temperature sensors, and internal temperature and workload data from those servers whose inlets are adjacent to the external sensors. Our prototype implementation abstracts away certain details and complexity for the sake of speed and simplicity. For example, we use CPU utilization as a proxy for workload. We felt this to be a reasonable simplification given that our CPUs are responsible for nearly over 80% of the server's power consumption.

The raw data for each server type comes from three sources: CPU utilization data, internal temperature data from kernel interfaces, and external temperature sensor networks. Once the raw data was collected from all three sources, we synchronized internal and external data. Internal data for which the corresponding external data was "stale" (over 60 seconds old) was discarded. Finally, we selected a random subset of 10 servers for five-fold cross-validation.

In addition to varying the number of recent epochs we use as input, we vary the number of workload epochs and internal sensor epochs independently. This separation allows us to examine whether workload or internal sensors play a more significant role in constructing accurate models. While not exhaustive, this parameter space exploration comprises 810 unique models. Using general full factorial design analysis, we can identify which parameters have a significant effect, and for which parameters we can simply select a "reasonable" value.

ID	Parameter	% of Variation
A	Epoch Length (s)	0.34
B	Workload Epochs	0.14
C	Sensor Epochs	0.05
D	Target MSE	1.55
E	Sigmoid Slope	0.00
F	FFCV Index	**74.49**

Table 2: Percent of variance in accuracy attributable to first-order effects. Other than variance among FFCV experiments, only the target MSE accounts for any measurable variation.

Certain FANN implementation parameters were constant for all experiments, including the maximum number of training iterations (10^5), the number of hidden layers (two), and the size of the hidden layers (twice the number of neurons as the input layer).

5.1 Preliminary Experiments

Our data set comes from a corporate data center containing several hundred DL360 servers. We identified a dozen servers with external temperature sensors situated directly in front of their front air inlet panels. For a period of 45 hours, we collected CPU data at 1 second granularities, internal temperature data at 5 second granularities, and external temperature data when provided by the external sensor infrastructure.

At the time of observation the data center was in heavy use running large computational batch jobs. This provided for moderate variation in both processor utilization and ambient air temperature. Server inlet temperatures varied between 20°C and 28°C.

Table 2 charts the average sum-of-squared error (SSE) between the inferences made by our models and the actual ambient air temperatures, and quantifies the first-order effects of each parameter on model accuracy. Again, while first-order effects do not capture any interactions between parameters they describe over three-quarters of the variance regarding inference accuracy.

Variance in inference accuracy is dominated by one factor: the FFCV sub-experiment. All other first-order factors combined account for approximately 2% of the variance in inference accuracy. However, this

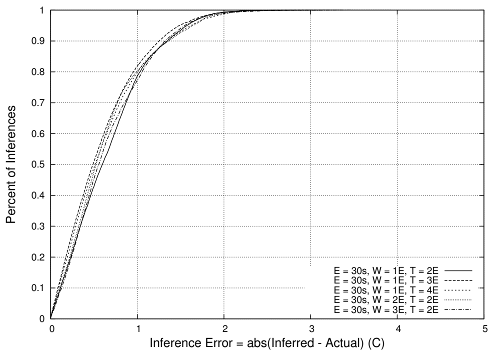

Figure 2: The CDF of inference error for five different models on the HP DL360; the epoch time is 30 seconds. In each case over 80% of the inferences are within 1°C of the actual server inlet temperature.

may indicate that the range of values we selected were not sufficiently varied to reveal significant differences between them. It is worth noting, though, that combining the effects of certain parameter selections could have more significant implications on the accuracy of our inferences.

Finally, we graph the accuracy of our inferences. Figure 2 shows the CDF of our model's inference accuracy for 5 combinations of our parameters. The x-axis is the absolute value of the difference between the inferred value of ambient air temperature and the actual temperature. In each case, over 80% of the inferences are within 1°C, and over 95% of the inferences are within 1.5°C of the correct value.

6 Conclusion

Our approach leverages ongoing standardization in on-chip and on-board internal sensors and the rich set of tools available for workload instrumentation. We develop a model that uses internal sensor readings and workload utilization metrics to infer the external ambient air temperature at *every* server inlet in the data center, based on a one-time calibration on a few machines. In addition to the increased coverage, our results also has fairly high *accuracy.* Our analysis shows that the model is capable of inferring ambient air temperature within 1°C over 80% of the time.

Furthermore, our approach just needs us to deploy the model on the server and can easily be ported to any data center in a fairly small amount of time. Additionally, our approach addresses another key challenge with external sensors, namely the problem of synchronizing and correlating ambient temperature readings with the equivalent workload utilization metrics. This enables better integration into higher-level thermal control loops such as for reduced cooling costs and greater reliability.

References

[1] The Fast Artificial Neural Net Library, May 2005. `http://leenissen.dk/fann/`.

[2] D. Anderson, J. Dykes, and E. Riedel. More Than an Interface—SCSI vs. ATA. In *Proceedings of the 2nd Usenix Conference on File and Storage Technologies (FAST)*, San Francisco, CA, March 2003.

[3] D. J. Bradley, R. E. Harper, and S. W. Hunter. Workload-based Power Management for Parallel Computer Systems. *IBM Journal of Research and Development*, 47:703–718, 2003.

[4] 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)*, pages 103–116, October 2001.

[5] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Mobile Computing and Networking*, August 2000.

[6] J. Jung, B. Krishnamurthy, and M. Rabinovich. Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. In *In Proceedings of the 2002 International World Wide Web Conference*, pages 252–262, May 2002.

[7] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making Scheduling "Cool": Temperature-Aware Workload Placement in Data Centers. In *Proceedings of the 2005 USENIX Annual Technical Conference*, pages 61–74, April 2005.

[8] J. Mouton. Enabling the vision: Leading the architecture of the future. In *Keynote speech, Server Blade Summit*, 2004.

[9] C. D. Patel, C. E. Bash, R. Sharma, and M. Beitelmal. Smart Cooling of Data Centers. In *Proceedings of the Pacific RIM/ASME International Electronics Packaging Technical Conference and Exhibition (IPACK03)*, July 2003.

[10] K. Rajamani and C. Lefurgy. On Evaluating Request-Distribution Schemes for Saving Energy in Server Clusters. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software*, March 2003.

[11] R. R. Schmidt, E. E. Cruz, and M. K. Iyengar. Challenges of data center thermal management. *IBM JOURNAL OF RESEARCH AND DEVELOPMENT*, 49:709–723, 2005.

[12] 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, Jan 2005.