Weatherman: Automated, Online, and Predictive Thermal Mapping and Management for Data Centers

Justin Moore and Jeffrey S. Chase
Duke University Department of Computer Science
Durham, NC
{justin, chase}@cs.duke.edu

Parthasarathy Ranganathan
Hewlett-Packard Labs
Palo Alto, CA
partha.ranganathan@hp.com

Abstract— **Recent advances have demonstrated the potential benefits of coordinated management of thermal load in data centers, including reduced cooling costs and improved resistance to cooling system failures. A key unresolved obstacle to the practical implementation of thermal load management is the ability to predict the effects of workload distribution and cooling configurations on temperatures within a data center enclosure. The interactions between workload, cooling, and temperature are dependent on complex factors that are unique to each data center, including physical room layout, hardware power consumption, and cooling capacity; this dictates an approach that formulates management policies for each data center based on these properties.**

We propose and evaluate a simple, flexible method to infer a detailed model of thermal behavior within a data center from a stream of instrumentation data. This data — taken during normal data center operation — includes continuous readings taken from external temperature sensors, server instrumentation, and computer room air conditioning units. Experimental results from a representative data center show that automatic thermal mapping can predict accurately the heat distribution resulting from a given workload distribution and cooling configuration, thereby removing the need for static or manual configuration of thermal load management systems. We also demonstrate how our approach adapts to preserve accuracy across changes to cluster attributes that affect thermal behavior — such as cooling settings, workload distribution, and power consumption.

I. Introduction

Power consumption and heat management have emerged as key design challenges in creating new data center architectures. In addition to the increased cooling costs resulting from larger installations, heat dissipation can also adversely impact system reliability and availability. This problem will be exacerbated by ongoing trends towards greater consolidation and increased density [1], [2]. For example, popular "blade" systems pack more computing in the same volume, increasing heat densities by up to a factor of seven in the next few years [2].

The growing importance of this problem has led to the development of several thermal management solutions, both at the facilities and at the IT (systems) level. Facilities-level solutions include the development of better cooling solutions both at the component level (e.g., better air conditioning units) and at the data center level (e.g., aisle layout to improve cooling efficiency [3]). More recently, Patel et al. [4] have shown that fine-grained cold air delivery based on a detailed thermal profile of the data center can provide significant additional efficiency improvements. Similarly, at the systems level, past work has focused on power consumption and heat dissipation at the component level (e.g., at the front-end servers [5], [6]) and at the data center level (e.g., power-aware resource provisioning [7], [8], [6], [9]). More recent work has focused on fine-grained thermal control through temperature-aware resource provisioning [10] and temperature-aware resource throttling [11].

A key challenge in these and other future optimizations is the need to *predict* the *heat profile*, the temperature at individual locations throughout the data center. This is determined by the *thermal topology* of the data center. The thermal topology describes how and where heat flows through a data center and determines the heat profile for a given configuration. Once the heat profile is known, it can be used to determine the properties of that configuration; this includes cooling costs, cooling efficiency, long-term component reliability, and the number of individual servers in danger of triggering their internal thermal "kill" switch (among others).

However, understanding the thermal topology and predicting the heat profile is often complex and non-intuitive. The thermal topology is a function of several factors, including the physical topology of the room, the distribution of cooling, and the heat generated by the individual servers (we discuss these further in Section II). Furthermore, many of these parameters change continuously during the day-to-day operation of the data center and have non-linear interactions with the thermal topology. Past work on thermal optimizations laid the foundation for thermal management through the use of simple methods. These include using either proxies or heuristics — i.e., using the overall power consumption [7] or a single-point temperature [12] — to characterize the "goodness" of the solution, running time-consuming thermo-dynamics simulations, or conducting elaborate calibration experiments — requiring the entire data center to be taken offline — to evaluate the heat profile for each configuration [10]. However, as optimizations focus on power and cooling control at a finer granularity [1],

1-4244-0175-5/06/$20.00 ©2006 IEEE.

it becomes more important to formulate better models of the data center thermal topology, predicting the heat profile in real time and at low cost.

Our work addresses this challenge by developing *automated, online, predictive thermal management for data centers*. We make two key contributions:

We demonstrate *automated modeling of data center thermal topology.* Weatherman, our proof-of-concept prototype, uses standard machine learning techniques to show that it is possible to learn and predict the complexities of the thermal topology of a 1000-plus-node data center using measurements from day-to-day operations. The experimental results show that our approach is accurate. Over 90% of our predictions are within 0.87°C of the actual temperature, while achieving more than a 10,000-fold improvement in running time.

Second, we discuss *the benefits of an online approach to predicting the heat profile for a given data center configuration.* In particular, we focus on a temperature-aware resource provisioning algorithm that uses coordinate-space search in conjunction with our model. Our algorithm performs as well as the previously published best algorithm — reducing cooling costs by 13% to 25% during moderate to heavy data center utilization — while eliminating the "offline" requirements of the prior work. In addition to cost savings, our model enables a quantitative comparison between proposed workload distributions, giving the data center owner greater flexibility to optimize operations using multiple metrics.

Section II further discusses the challenges with modeling thermal topologies and past work. Section III presents a formal problem statement, while Section IV describes our approach using machine learning methods. Section V discusses the benefits from our model and its use in temperature-aware workload distribution. Section VI concludes the paper.

II. Motivation and Related Work

The goal of this work is to explore the feasibility of creating a model that predicts how facilities components — such as computer room air conditioning (CRAC) units, the physical layout of the data center, and IT components — will affect the heat profile of a data center. An accurate thermal topology of a data center can:

Enable holistic IT-facilities scheduling. One of the significant advantages of a thermal topology model is the ability to to quantify the total costs associated with a configuration. Being able to measure the absolute differences in the costs, as opposed to a simple relative ordering of configurations, can help when considering holistic QoS-aware IT/facilities optimizations [13] targeted at the total cost of ownership.

Increase hardware reliability. A recent study [3] indicated that in order to avoid thermal redlining, a typical server should have the air temperature at its front inlets be in the range of 20°C – 30°C. Every 10°C increase over 21°C decreases the long-term reliability of electronics, particularly disk drives, by 50% [3], [14], [15].

Decrease cooling costs. In a 30,000 ft^2 data center with 1000 standard computing racks, each consuming 10 kW, the initial cost of purchasing and installing the CRAC units is $2 – $5 million; with an average electricity cost of $100/MWhr, the annual costs for cooling alone are $4 – $8 million [4].

Decrease response times to transients and emergencies. Data center conditions can change rapidly. In data center with high heat densities, severe transient conditions — such as those caused by utilization spikes [16], [17] or cooling failure [10] — can result in disruptive downtimes in a matter of minutes or seconds.

Increase compaction and improve operational efficiencies. A high ratio of cooling power to compute power limits the compaction and consolidation possible in data centers, correspondingly increasing the management costs.

A. Challenges

At a high level, we are attempting to model the injection, flow, and extraction of hot air. The main obstacles to achieving this goal are the non-intuitive nature of heat flow and non-linear equations governing certain aspects of heat transfer. Prior work demonstrated how the thermal effects of increased server utilization could be spatially uncorrelated with that server or group of servers [10]. Additionally, while some parameters to fluid mechanics equations have linear effects — such as temperature and heat — other parameters have non-linear effects — including air velocity and buoyancy.

If we can enumerate the primary factors that serve as inputs (I) to the thermal topology of a data center (T) we can model the effects of those factors on the resulting thermal map (M):

$$M = T(I)$$

Therefore, a robust model that accurately describes all linear and non-linear thermal behavior within the data center can predict values of M for all values of I.

A primary challenge in characterizing the thermal topology is the variability of the numerous components in the data center. For example, the power distribution is influenced by the utilization pattern of the data center (which for most Internet workloads is quite noisy) as well as the application and resource usage characteristics of the workload. Several factors affect the air-flow in the data center, including unintentional obstructions to the air-flow from vents, open rack doors, fan or CRAC failure, etc. In addition, intentional variation to the cooling such as that proposed in [4] can also change the thermal topology. Second-order variations such as temperature-sensitive variations in power consumption and air-flow properties as well variation in the speeds of the fan and the associated variability in their heat dissipation adds other variability to the thermal topology.

It is possible to calculate the exact thermal topology model using three-dimensional numerical analysis solving for the laws of thermodynamics; these are at the heart of computational fluid dynamics (CFD) applications [18]. CFD solvers use discretization techniques to transform the partial differential equations into algebraic form, iterating over the equation solutions until it reaches a a suitable convergence level.

However, CFD models have several drawbacks that prevent their use in online management algorithms. Both the initial costs (model creation) and recurring costs (model execution) of a CFD approach can take hours or days, depending on the complexity of the data center model. The transformation from differential equations to algebraic forms leads to a set of partial differential equations that are highly coupled and non-linear. Our desired method would produce an accurate answer in an online fashion, rather than a perfect answer in an offline fashion.

B. Related Work

Past work on data center thermal management took a modular approach by addressing different classes of challenges separately. For example, several projects reduced data center cooling costs using a variety of approaches, such as optimizing cooling delivery [4], minimizing global power consumptions [19], [7], [20], and efficient heat distribution [3], [10], [8], [21]. Each of these methods approaches the problem heuristically. Rather than calculate the complete thermal topology of the data center, they select a data center property that is associated with an efficient thermal topology — such as low server power consumption, a lower CRAC return temperature, a uniform exhaust profile, or minimal mixing of hot and cold air — and alter the power or cooling profiles to optimize along these specific metrics.

These selective approaches have obvious benefits and drawbacks. The primary benefits are efficiency and simplicity, both in the time required to create a model of how power and cooling profiles affect the metric, and the accuracy of predicting metric values given a power and cooling profile. For example, our work in temperature-aware workload placement [10] divides the data center into "pods" and measures the global level of mixing between cold air and the hot air coming from servers in each pod. Even though this approach is agnostic as to the *location* of such mixing, it enables significant data center cooling cost savings.

The primary drawback, though, is an incomplete view of the thermal topology. These approaches are state of the art heuristic methods, and are feasible because they assume a portion of the power or cooling profile is fixed, or they make simplifying assumptions regarding secondary effects. For example, [10] assumes that, as the number of utilized servers increases, the temperature of the air supplied by the CRAC units will change uniformly; that is, all CRAC units supply cold air at the same temperature, and that temperature changes simultaneously on all units. Any changes to individual supply temperatures or the fan speed of any CRAC unit will alter the amount of mixing that occurs between the incoming cold air and the hot exhaust from servers, thereby changing the relative efficiencies of the servers. The workload distribution algorithm would need a complete new set of input data for its heuristic.

The other consequence of the incomplete thermal topology is that, while these prior approaches can help determine the *qualitative* benefits across multiple configurations (such as a ranked list of servers, ordered by how much they increase cooling costs), they cannot quantify the final effects of their decisions. In some optimizations it may be beneficial to choose a configuration with slightly inferior thermal properties so that a different metric (e.g., locality, network congestion) can be optimized for a better overall total cost of ownership.

III. Problem Formulation

Before selecting an appropriate technique to model data center thermal topology, we must formalize our problem statement. In this section we define the relevant *model parameters*; that is parameters that are necessary to construct any thermal topology, independent of the method chosen to implement that model. In Section IV we discuss the *implementation parameters* that are specific to our prototype.

A. Problem Statement

In Section II-A we described the thermal topology as being a function by which we predict the thermal map that will result from a given set of input factors:

$$M = T(I)$$

In order to formulate a problem statement, we must enumerate the variables in I that affect the thermal topology, and what instrumentation values are sufficient to provide a useful M.

There are three primary input factors:

Workload distribution (W), which includes utilization data for any hardware that produces measurable amounts of heat. Servers, storage, network switches, and other hardware falls into this category. In practice, we can obtain this data — including, but not limited to, CPU utilization, disk I/O rates and rotate speed, memory I/O rates, and network activity — from any number of available instrumentation infrastructures.

Cooling configuration (C) of the room, including the number and distribution of CRAC units, their air flow velocity, and the temperature of the air they supply to the data center. This configuration also includes non-CRAC factors that affect air flow in a data center, including fan speeds of the servers.

Physical topology (P). The physical topology consists of the objects in the room, including the locations of server racks, walls, doors, and slotted floor tiles.

We represent each of these factors as a one-dimensional array of values. For example, if there are X servers in the data center, we represent W as

$$W = [W_0 W_1 \dots W_X]$$

We make a similar generalization for the thermal map, specifying a set of instrumentation values that provide an accurate representation of the map. This results in our formal problem statement:

$$M = T(W, C, P)$$

The set of values contained in W, C, and P are the input to our function, and the set of values contained in M are our output.

IV. WEATHERMAN

This section discusses the specific input parameters, mathematical methods, source data, and software systems used to implement Weatherman, our prototype model construction application.

A. Data Collection

The first step in implementing Weatherman 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 aggregation and correlation. Server utilization is available through any number of standard monitoring infrastructures [22], [23], [24]. CRAC data — such as fan speeds and air temperature — is available through instrumentation infrastructures on modern systems [25]. The output data — sensors that measure ambient air temperature — can be collected through any number of available hardware and software infrastructures [26], [25].

Prior to model construction, we tag the readings with metadata to indicate the object of origin. For input data, this will be the server or CRAC from which the readings came. The server of origin for output data will come from the external temperature sensor that is located directly in front of that server.

B. Machine Learning

Exact solutions using CFD methods are too complex and time-consuming for online scheduling. Therefore, we turn to methods that provide approximate solutions. The field of machine learning contains several methods for finding approximate solutions of complex problems with large data sets. Additionally, there are several "off-the-shelf" machine learning development libraries, enabling us to leverage these techniques rapidly. In essence, our thermal topology model "learns" how the values of dynamic input parameters affect heat flow, allowing us to predict the heat profile that results from a given power and cooling profile.

The first step in using machine learning is identifying the necessary properties of our thermal topology, and using these properties to select an effective learning technique. Our technique must be capable of producing outputs that fall within a continuous range and can be the product of non-linear relationships between the inputs; these criteria rule out classification techniques such as decision trees, tree induction algorithms, and propositional learning systems.

Neural nets, on the other hand, contain all the necessary properties [27], [28]. Additionally, they present a reasonable analogy to our thermal topology, as input values "flow" through the net to the output values in much the same way that air flows through our data center. 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 air injection and observed temperature depends on the structure of the data center.

In Weatherman, the data sets are pairs of power profiles and heat profiles, taken while the data center is at a temporary steady-state. The strength of this approach is that it allows us to add measurements to our training set during normal operation of the data center. Furthermore, the more often we operate at a given utilization level, and the more unique workload distributions we capture at that utilization level, the better the model "learns" the thermal topology for that utilization level. For example, a data center that hosts long-running batch jobs using a scheduler that deploys jobs randomly can collect a significant number of unique power and heat profiles. In turn, the model uses these unique pairings to predict heat profiles for all possible power profiles without the need to "see" every possible unique power profile.

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 are capable of producing models that have the properties we desire in our solution. Ultimately, newer methods — such as support vector machines — may produce results superior to those possible with neural networks. However, this work should be seen as only the first attempt to merge thermal mapping with machine learning techniques.

C. 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 [29]. FANN implements standard neural net training and execution functions, allowing us to focus on exploring effective methods of constructing our models rather than routine implementation details.

Weatherman leverages the properties of neural nets to predict how heat is generated and flows within the data center. For a data center X workload parameters, Y cooling settings, and Z room layout measurements, there will be $N = X + Y + Z$ inputs to our model. The output of our model will be the M measurements that comprise our thermal map.

Between the input layer and the output layer, there are L *internal* or *hidden* layers. Each layer contains a set of elements known as *neurons*. Each neuron i accepts N_i inputs from the previous layer, applies a weighting factor $w_{i,a}$ to each input x_a, and uses the sum of the weighted inputs as the x-value for its activation function, g. The result of this function, y_i is passed to neurons in the next layer.

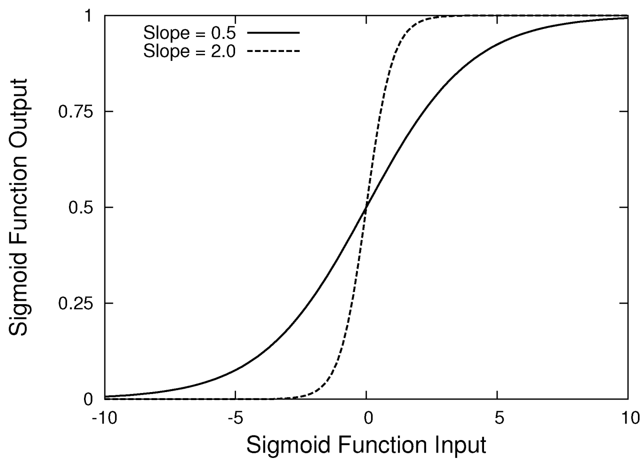

Fig. 1. Effects of steepness on two sigmoid functions: s_1 = 0.5, s_2 = 2.0. Smaller s-values provide a greater ability to make subtle distinctions during training, but can lead to over-training.

$$y_i = g\Big(\sum_{a=0}^{N_i} w_{i,a} \cdot x_a\Big)$$

Of the three activation functions implemented in the FANN library, the sigmoid activation function meets the necessary criteria. It only allows positive output values from neurons and outputs contiguous values.

$$g(x) = \frac{1}{1+e^{-(x \cdot s)}}$$

The sigmoid parameter s controls the *steepness* of the output slope, and is an implementation parameter of interest. Figure 1 shows the shape of two sigmoid functions with different s-parameters. An overly steep sigmoid function requires precise inputs at all stages of the neural net to produce accurate outputs; small errors grow as they pass through the network, producing incorrect outputs. However, a sigmoid function that is "flat" may result in an overly trained network. In other words, it can make accurate predictions for inputs similar to previously-seen data, but is not general enough to provide accurate answers for new input sets.

D. Preprocessing, Training, Testing, and Validation

The first stage in constructing our model is preprocessing our input and output data sets. Given that output values from the net will be in the range $[0, 1]$ — due to the sigmoid function — we scale all input and output values to fall within this range. This provides consistency between input and output data, and allows the model to predict a wide range of thermal map temperatures.

Next, we select a set of values for our model and implementation parameters and construct a neural net by calculating the weights for each input to each neuron. This phase of creating a single neural net is known as *training* the network. The training phase involves providing a set of inputs and outputs, and adjusting the weights to minimize the *mean square error* (MSE) between the predicted outputs and actual outputs over the entire set of training data.

Training is, in essence, an optimization problem that minimizes the MSE. It can leverage techniques such as genetic algorithms, simulated annealing, or back-propagation. The back-propagation algorithm in FANN works by calculating the MSE at the output neurons, and adjusting the weights through the layers back to the input neurons. FANN trains on each individual input/output pair and performing back-propagation sequentially, rather than training on the combined data. This method results in faster training times, but makes the ordering of the data sets significant.

This iterative training process continues until the MSE reaches a user-defined minimum threshold or the training process has executed a specified number of iterations. Therefore, MSE is an implementation parameter of interest.

The third stage of model construction — and the second stage in constructing a single neural net — is *testing* the network. Testing involves using the neural net to predict the outputs for a given set of inputs that were not present in the training data. Testing examines to what extent the neural net is generally applicable, and that the training session did not create a net that is overly trained to inputs it has already seen.

Finally, we quantify the suitability of a given set of model and implementation parameters by calculating the *sum of squared error* (SSE) across multiple neural nets. A small SSE indicates the model and implementation parameters generate suitably accurate models. Using standard analysis of variance techniques, we can isolate the effects of parameter selection on the accuracy of our models.

V. Results

We now present the results using Weatherman to learn a thermal topology. We describe the training process, and demonstrate Weatherman's ability to predict the heat profile resulting from new workload distributions.

A. Data Center Simulations

We study a typical medium-sized data center, as shown in Figure 2. The data center contains four rows of seven racks, containing a total of 1120 servers. The data center has alternating "hot" and "cold" aisles. The cold aisles, B and D, have vented floor tiles that direct cold air upward towards the server inlets. The servers eject hot air into the remaining aisles: A, C, and E. The data center also contains four CRAC units. Each CRAC pushes air chilled to 15°C into the plenum at a rate of 10,000 $\frac{ft^3}{min}$. The CRAC fans consume 10 kW each.

Each 1U server has a measured power consumption of 150W when idle and 285W with both CPUs at 100% utilization. The total power consumed and heat generated by the data center is 168 kW while idle and 319.2 kW at full utilization. Percent utilization is measured as the number of machines that are running a workload. For example, when 672 of the 1120 servers are using both their CPUs at 100% and the other 448 are idle, the data center is at 60% utilization.

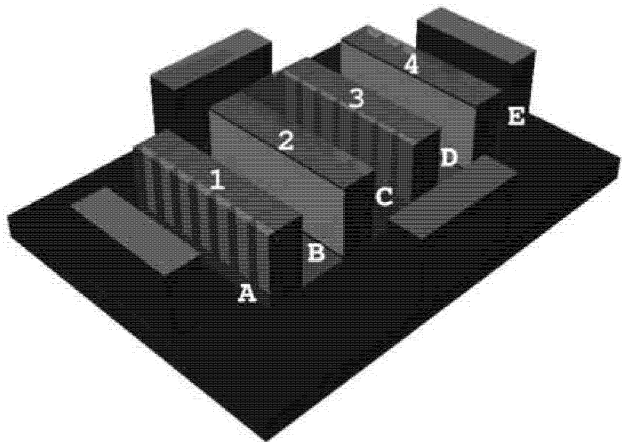

Fig. 2. Data center layout, containing 1120 servers in four rows of seven racks. The racks are arranged in a standard hot-aisle/cold-aisle configuration [3]. Four CRAC units push cold air into a plenum, which then enters the room through floor vents in aisles B and D. Servers eject hot air into aisles A, C, and E.

Order	% of Variance
ABCD	19.2
ABD	10.2
AB	10.0
BCD	8.7
ABC	8.7
ACD	7.8
CD	5.7
AD	5.3
BC	5.2
BD	5.2
AC	4.5
B	3.7
D	3.2
C	1.8
A	0.8

TABLE II

BREAKDOWN OF VARIANCE IN MODEL ACCURACY BY PARAMETER INTERACTIONS; SEE TABLE I FOR PARAMETER NAMES AND VALUES.

ID	Parameter	P_1	P_2	P_3
A	Block Size	4	10	20
B	KW Scale	200	300	400
C	Target MSE	10^{-5}	$5 \cdot 10^{-5}$	$2.5 \cdot 10^{-4}$
D	Sigmoid	$1 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$

TABLE I

THE LIST OF MODEL PARAMETERS (A) AND IMPLEMENTATION PARAMETERS (B, C AND D), AND THE LIST OF POSSIBLE VALUES WE ASSIGN TO THEM DURING TRAINING.

Ideally, to validate accuracy, we would like to compare the heat profile from our model with that from instrumentation of a real data center. Given the costs and difficulties of instrumenting and performing experiments on this sized data center, we instead used the CFD approach discussed earlier with Flovent [18], a commercially available simulator. At the conclusion of each simulation, Flovent provides the inlet and exhaust temperature for each object in the data center. Previous work validated the accuracy of Flovent-based simulations with experiments on a real data center [12].

B. Configuration

The first step in constructing our model is to obtain the data for the training sets. From previous work [10], [12], we had a library of nearly 360 Flovent runs which tested over 25 unique workload distribution algorithms at multiple levels of data center utilization. We selected 75 simulations as representing how an actual data center might distribute batch workloads of varying sizes. Data from the remaining experiments was used to test the accuracy of our models.

We selected a neural net configuration after discussing the system requirements with an AI researcher. All the neural nets we trained consisted of four layers: one input layer, two hidden layers, and one output layer. This configuration allows the nets to capture the complex, non-linear thermal relationships inherent in the data center environment. The size of the input layer is dependent on the number of servers, the block size (parameter A), and the number of CRAC units. The two hidden layers are each twice the size of the input layer. Finally, the output layer has a neuron for each temperature we would like to predict. In our model, this translates to one neuron for each block.

Given that a model which represented each server with a single input would be too large for our four-layer net — it would contain 6.3 trillion neurons — we divided the servers into contiguous blocks. The value of each input neuron was the sum of the power consumed by all servers in that block. We then divide the kilowatts consumed by each block by a scaling factor, as described in Section IV-C. This simplification is based on the assumption that the heat generated by adjacent servers — a 1U server is 1.75" tall — are not significant compared to other data center properties.

Table I specifies the model and implementation parameters we explored in creating Weatherman models; in all, we trained 81 models. For each parameter we attempted to select one value that was overly aggressive and likely to result in an overly-trained net, one value that would result in a significantly less accurate net, and one "ideal" target value. If our assumptions regarding target accuracy, block size, or scaling were invalid, an analysis of the results would indicate a statistically significant difference in the accuracy of the nets that were trained using those parameters.

C. Accuracy

Table II shows the sensitivity of our models to changes in parameter values. Changing an individual parameter has little effect on the accuracy of a model; changing multiple parameters in concert accounts for the most variance in model accuracy. For example, simultaneous changes in block size (A), the target MSE (C) and the sigmoid exponent (D) account for 7.8% of the variance in model accuracy.

The model we ultimately selected has a 4U block size, a

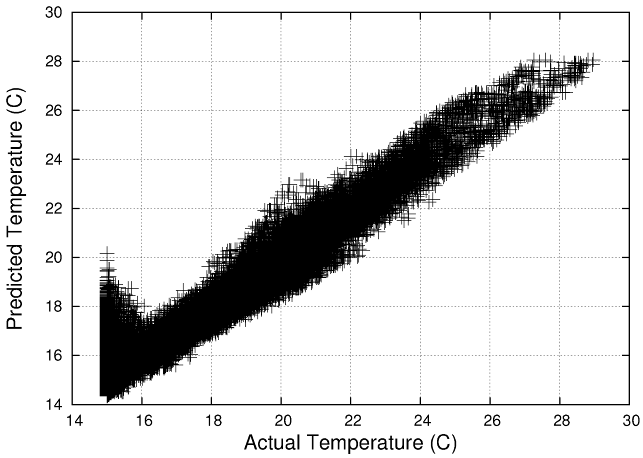

Fig. 3. Scatter-plot of predicted values versus actual values. A perfect model would create a straight line with a slope of one. Our predictions are accurate across the 15°C range of values.

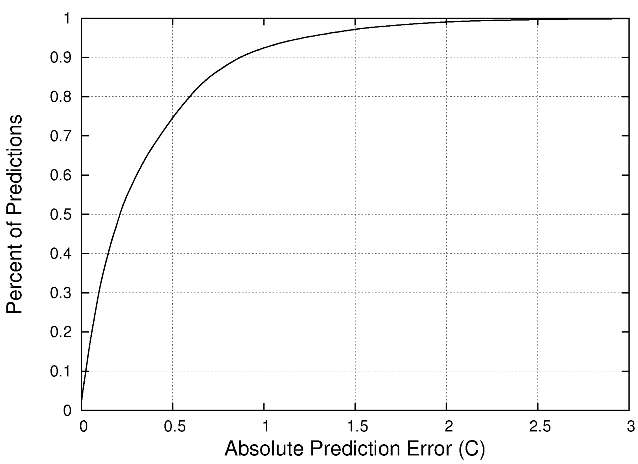

Fig. 4. CDF of the error between actual and predicted values. Over 90% of predictions are accurate within 0.87°C; the median error is 0.22°C.

200 KW scaling value, and small MSE and sigmoid values. This produces a model that is accurate and able to learn how subtle differences in the input values affects the thermal profile. Figure 3 shows a scatter plot of predicted temperature value distribution versus the actual distribution for our 280 test experiments (a total of 313,600 data points), while Figure 4 shows a CDF of the accuracy of our predictions. Over 75% of our predictions are within 0.5°C, and 92% are within 1.0°C.

Given that the accuracy of most hardware temperature sensors is within 1.0°C [26], this demonstrates that it is possible to construct thermal topology models whose accuracy is within the margin of error for off-the-shelf temperature sensors. To our knowledge, ours is the first work to prove that such an approach is feasible, using data available from day-to-day instrumentation.

D. Workload Placement

Here we describe one possible use of our thermal topology: temperature-aware workload placement. We provide a brief background in the thermodynamics of cooling cycles and how we calculate cooling costs. We then discuss how to leverage the thermal topology in selecting servers that lead to reduced cooling costs.

1) Thermodynamics: The efficiency of a cooling cycle is quantified by a *Coefficient of Performance* (COP). The COP is the ratio of heat removed (Q) to the amount of work necessary (W) to remove that heat. Conversely, a larger COP indicates a more efficient process, requiring less work to remove a constant amount of heat.

$$W = \frac{Q}{COP}$$

However, the COP for a cooling cycle is not constant, increasing with the temperature of the air the CRAC unit pushes into the plenum. By raising the temperature of the air supplied to the room, we operate the CRAC units more efficiently. For example, if air returns to the CRAC unit at 20°C and we remove 10 kW of heat, cooling that air to 15°C, we expend 5.26 kW. However, if we raise the plenum supply temperature to 20°C, everything in the data center warms by 5°C. Cooling the same volume of air to 20°C removes the same 10 kW of heat, but only expends 3.23 kW; this is a power savings of almost 40%.

For a thorough discussion of the relevant thermodynamics, see [10].

2) Calculating Cooling Costs: We calculate the cooling costs for each run based on a maximum safe server inlet temperature, T^{in}_{safe}, of 25°C, and the maximum observed server inlet temperature, T^{in}_{max}. We adjust the CRAC supply temperature, T_{sup}, by T_{adj}, where

$$T_{adj} = T^{in}_{safe} - T^{in}_{max}$$

If T_{adj} is negative, it indicates that a server inlet exceeds our maximum safe temperature. In response, we lower T_{sup} to bring the servers back below the system redline level. Our cooling costs can be calculated as

$$C = \frac{Q}{COP(T = T_{sup} + T_{adj})} + P_{fan}$$

where Q is the amount of power the servers consume, $COP(T = T_{sup} + T_{adj})$ is our COP at $T_{sup} + T_{adj}$, and P_{fan} is the total power required to drive the CRAC fans.

3) Baseline Algorithms: We study three workload distribution algorithms as points of comparison to our thermal-topology-based approach. UNIFORMWORKLOAD takes the total power consumption of the N servers in the data center, and assigns $\frac{1}{N}^{th}$ of that power to each server. UNIFORM-WORKLOAD emulates the behavior of a random scheduler over time, as each server is equally likely to use the same amount of power over a long enough time window.

MINHR and MAXHR are the best and worst workload distributions as described in [10]. They attempt to minimize and maximize, respectively, the amount of hot exhaust air that mixes with the cold air streams coming from the CRAC units.

4) Using Weatherman for Workload Placement: The difficulty in using the thermal topology to select a desirable set of servers for a given data center utilization is that we are attempting to "invert" the topology. Instead of using a known power profile to calculate a heat profile, we are attempting to discover an unknown power profile that has a desirable heat profile. For any workload that uses N of the M servers in the data center, there are $\binom{M}{N}$ possible unique power profiles; for example, even if we constrain ourselves to use servers in blocks of five — all five are either used or idle — there are over 10^{66} possible unique power profiles at 50% utilization. We are faced with a new challenge, in that we must use a heuristic to search this space to locate a reasonable power profile.

The method we selected is a *coordinate-space search*, a two-stage workload placement heuristic. In the first stage we calculate the cooling costs at the initial state; depending on the data center owner's policy, this could involve having all machine sitting on but idle, or having all machines turned off. We then calculate the cooling costs at this initial state.

The second stage consists of selecting on which servers we will place our workload. We maintain two lists: an *active list* and an *idle list*. The active list contains the current set of servers we will use and is initially empty, while the idle list initially contains all servers. We operate at the granularity of a server block, as defined in Table I. In each iteration of our search, we determine which block of servers in the idle list would — if utilized — result in the smallest increase in cooling costs. We can perform each iteration through a simple linear scan of the current servers in the idle list. We then add the selected block of servers to the active list and begin a new iteration. The search terminates when the active list contains enough servers to run our workload.

Our heuristic has several desirable properties. First, its runtime is $O(N \cdot M)$, significantly smaller than $\binom{M}{N}$. Second, it is deterministic; this allows us to preprocess the results for a set of discrete utilization levels. Third, it creates a ranked list of servers. This simplifies the process of integrating the results of our workload placement algorithm with existing tools, such as batch queues.

5) Cooling Costs: Figures 5 and 6 demonstrate the effectiveness of using thermal topology for data center workload placement. Note that workload distributions based on predictions using a thermal topology reduce hot air recirculation as much as — if not more so — than MINHR, which uses extensive a priori knowledge specifically for the purpose of reducing such recirculation. Furthermore, our distribution algorithm results in cooling costs comparable to those produced by MINHR, and Weatherman achieves a 13% – 25% reduction in cooling costs over the UNIFORMWORKLOAD algorithm.

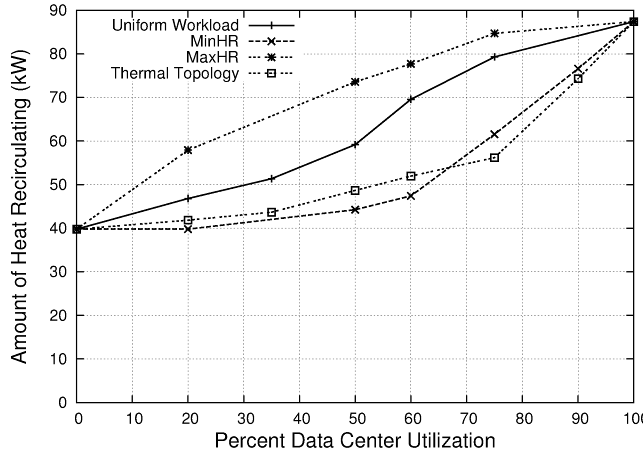

Fig. 5. Heat recirculation for our three baseline algorithms and our thermal-topology-based algorithm. Weatherman reduces the recirculation of hot air as well as, if not better than, the MINHR algorithm.

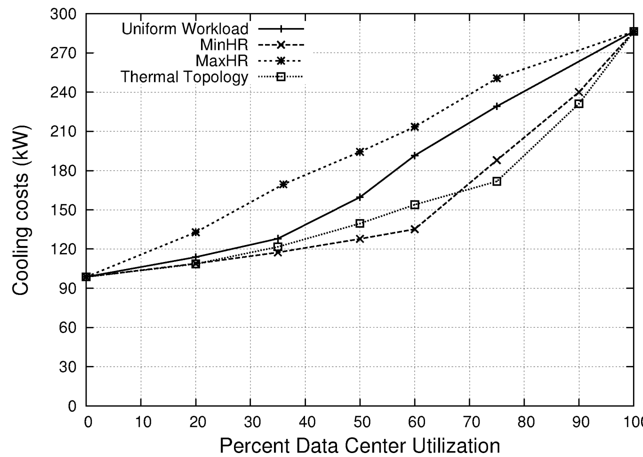

Fig. 6. Cooling costs for our three baseline algorithms and our thermal-topology-based algorithm. The Weatherman-based workload placement algorithm achieves savings comparable to the previous best in temperature-aware workload placement.

E. Discussion

1) Workload Placement Observations: The differences in cooling costs between MINHR and Weatherman-based workload placement are due to two primary considerations.

First, Weatherman is a generalized method for making quantitative predictions about data center conditions, while MINHR is a specialized and qualitative workload placement algorithm. The workload placement component of Weatherman is limited by the accuracy of its model and the search heuristic.

Our workload placement heuristic assumes the computer infrastructure only had two power states: idle and used. However, many data center management infrastructure components — such as networked power switches, blade control planes, and Wake-On-LAN-enabled Ethernet cards — allow us to consider "off" as another power state. Combining Weatherman with more sophisticated placement heuristics can leverage additional power states and enable greater power savings.

Furthermore, it is likely that superior modeling and search methods would yield better results. Yet even with off-the-shelf neural net software and greedy search algorithms, Weatherman performs comparably to a specialized placement algorithm.

Second, these Weatherman models use less data — 75 simulations — than the source data for MINHR — 112 simulations. Adding training data that represents a broader range of workload placement combinations would produce better models, allowing Weatherman to predict a more accurate thermal map.

2) Instrumentation: The work discussed in this paper assumes an instrumentation infrastructure in the data center that can provide accurate temperature and power readings at a fine granularity. Given that the data used to construct the models we analyzed were from a CFD simulation, we had access to temperature readings with three decimal places at arbitrary locations within the model. These two temperature instrumentation properties — accuracy and extensive coverage — do not exist in current data centers, and would degrade the quality of source data available to construct Weatherman models.

While we are uncertain how the "noise" in the source data stream would degrade the quality of Weatherman models, we feel there are reasonable mitigating factors. Fundamentally, machine learning methods are useful in scenarios where either the source data or the relationships between data points are non-intuitive, complex, or not accurate 100% of the time. Current efforts in data center instrumentation aim to reduce the granularity of temperature sensor coverage by using machine learning techniques to infer the ambient temperature in front of a server. Temperature instrumentation, both off-the-shelf hardware sensors and the machine learning software solution, does not introduce a systematic bias in the source data. Additionally, more sophisticated modeling methods and increased training and test data could offset errors in the underlying instrumentation.

VI. CONCLUSION

Cooling and heat management are fast becoming the key limiters for emerging data center environments. As data centers grow during the foreseeable future, we must expand our understanding of their thermal properties beyond simple heuristics-based techniques.

In this paper we explore factors that motivate modeling the complete thermal topology of a data center. We demonstrate a simple method by which one may construct these models using existing instrumentation culled from the day-to-day operation of a representative data center. The software used to construct these models leverages simple, off-the-shelf modules. The resulting accuracy of these models — our predictions are within 1.0°C of actual values over 92% of the time — show that even an naive approach is capable of yielding accurate predictions. Finally, we demonstrate that simple heuristics to search the large space of possible workload distributions result in energy-efficient solutions. We were able to improve upon existing heuristic-based workload distribution algorithms that were oblivious to the thermal topology and based management decisions on the metric of global heat recirculation.

Though we demonstrate the benefits of using Weatherman to minimize cooling costs, our models are also applicable to scenarios such as graceful degradation under thermal emergencies. In these cases, thermal-topology-aware measures can improve the response to failures and emergencies. Similarly, the principles underlying our heuristics can be leveraged in the context of dynamic control algorithms.

Overall, our work demonstrates that it is possible to have accurate, automated, online, and cost-effective thermal topology prediction. Most importantly, we provide the ability to make quantitative predictions as to the results of workload distribution and cooling decisions. To the best of our knowledge, our work is the first to demonstrate this. Our results demonstrate that such models can be beneficial in a variety of ways including improving previously-proposed techniques as well as enabling new approaches to data center heat management. As the problem of heat management becomes more and more critical, we believe that these and more sophisticated models will be an integral part of future designs.

VII. ACKNOWLEDGMENTS

We thank Mark Fashing, Austin Eliazar, and Ron Parr of Duke University for providing background and insight into neural net parameter selection and configuration.

We also thank Chandrakant Patel, Ratnesh Sharma, Cullen Bash, Monem Beitmal, and Eric Wu of HP Labs for arranging and coordinating time to conduct Flovent experiments.

REFERENCES

[1] R. Bianchini and R. Rajamony, "Power and Energy Management for Server Systems," *IEEE Computer*, vol. 37, no. 11, pp. 68–74, 2004.

[2] P. Ranganathan and N. Jouppi, "Enterprise IT Trends and Implications on System Architecture Research," in *International Conference on High-Performance Computer Architectures*, February 2005.

[3] R. F. Sullivan, "Alternating Cold and Hot Aisles Provides More Reliable Cooling for Server Farms," in *Uptime Institute*, 2000.

[4] 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.

[5] M. Elnozahy, M. Kistler, and R. Rajamony, "Energy Conservation Policies for Web Servers," in *In Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems*, March 2003.

[6] W. Felter, K. Rajamani, C. Rusu, and T. Keller, "A Performance-Conserving Approach for Reducing Peak Power Consumption in Server Systems," in *Proceedings of the 19th ACM International Conference on Supercomputing*, June 2005.

[7] 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, pp. 103–116.

[8] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath, "Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems," in *Proceedings of the Workshop on Compilers and Operating Systems for Low Power*, September 2001.

[9] V. Sharma, A. Thomas, T. Abdelzaher, K. Skadron, and Z. Lu, "Power-Aware QoS Management in Web Servers," in *24th International Real-Time Systems Symposium*, December 2003, pp. 63–72.

[10] 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*, April 2005, pp. 61–74.

[11] A. Weissel and F. Bellosa, "Dynamic Thermal Management for Distributed Systems," in *Proceedings of the First Workshop on Temperature-Aware Computing Systems (TACS)*, June 2004.

[12] R. K. Sharma, C. L. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase, "Balance of Power: Dynamic Thermal Management for Internet Data Centers," *IEEE Internet Computing*, vol. 9, no. 1, pp. 42–49, Jan 2005.

[13] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing Server Energy and Operational Costs in Hosting Centers," in *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, June 2005.

[14] 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.

[15] G. Cole, "Estimating Drive Reliability in Desktop Computers and Consumer Electronics," in *Technology Paper TP-338.1, Seagate Technology*, November 2000.

[16] M. Arlitt and T. Jin, "Workload characterization of the 1998 world cup web site," HP Research Labs, Tech. Rep. HPL-1999-35R1, September 1999.

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

[18] *Flovent version 2.1*, Flomerics Ltd., 81 Bridge Road, Hampton Court, Surrey, KT8 9HH, England, 1999.

[19] 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.

[20] H. Zeng, X. Fan, C. Ellis, A. Lebeck, and A. Vahdat, "ECOSystem: Managing Energy as a First Class Operating System Resource," in *Proceedings of Architectural Support for Programming Languages and Operating Systems*, October 2002.

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

[22] S. Godard, *SYSSTAT utilities home page*, November 2005, http://perso.wanadoo.fr/sebastien.godard/.

[23] F. D. Sacerdoti, M. J. Katz, M. L. Massie, and D. E. Culler, "Wide Area Cluster Monitoring with Ganglia," in *Proceedings of the IEEE Cluster 2003 Conference*, Hong Kong, 2003.

[24] *Hardware Monitoring for Linux*, LM Sensors Development Team, November 2005, http://secure.netroedge.com/∼lm78/.

[25] *OLE for Process Control Overview – Version 1.0*, OPC Foundation, October 1998, http://www.opcfoundation.org/.

[26] *DS1822 Econo 1-Wire Digital Thermometer*, Dallas Semiconductor, November 2005, http://www.maxim-ic.com/quick_view2.cfm/qv_pk/2795.

[27] J. S. Judd, "Learning in neural networks," *Proc. First ACM Workshop on Computational Learning Theory*, August 1988.

[28] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Transactions on Acoustics, Speech and Signal Processing*, pp. 4–22, April 1987.

[29] *The Fast Artificial Neural Net Library*, University of Copenhagen Department of Computer Science, May 2005, http://leenissen.dk/fann/.