A Sensor System for High-Fidelity Temperature Distribution Forecasting in Data Centers

JINZHU CHEN, Michigan State University
RUI TAN, Michigan State University and Advanced Digital Sciences Center, Illinois at Singapore,
YU WANG and GUOLIANG XING, Michigan State University
XIAORUI WANG and XIAODONG WANG, Ohio State University
BILL PUNCH and DIRK COLBRY, Michigan State University

Data centers have become a critical computing infrastructure in the era of cloud computing. Temperature monitoring and forecasting are essential for preventing server shutdowns because of overheating and improving a data center's energy efficiency. This article presents a novel cyber-physical approach for temperature forecasting in data centers, one that integrates Computational Fluid Dynamics (CFD) modeling, *in situ* wireless sensing, and real-time data-driven prediction. To ensure forecasting fidelity, we leverage the realistic physical thermodynamic models of CFD to generate transient temperature distribution and calibrate it using sensor feedback. Both simulated temperature distribution and sensor measurements are then used to train a real-time prediction algorithm. As a result, our approach reduces not only the computational complexity of online temperature modeling and prediction, but also the number of deployed sensors, which enables a portable, noninvasive thermal monitoring solution that does not rely on the infrastructure of a monitored data center. We extensively evaluated the proposed system on a rack of 15 servers and a testbed of five racks and 229 servers in a small-scale production data center. Our results show that our system can predict the temperature evolution of servers with highly dynamic workloads at an average error of 0.52°C, within a duration up to 10 minutes. Moreover, our approach can reduce the required number of sensors by 67% while maintaining desirable prediction fidelity.

Categories and Subject Descriptors: C.3 [**Special-Purpose and Application-Based Systems**]: Real-Time and Embedded Systems; C.4 [**Performance of Systems**]: Measurement Techniques, Modeling Techniques

General Terms: Experimentation, Design, Measurement, Performance

Additional Key Words and Phrases: Data center, cyber-physical system, wireless sensor network, temperature prediction, computational fluid dynamics

---

Part of this work was published in IEEE RTSS 2012.
This work is supported in part by the U.S. National Science Foundation under grants CNS-0954039 (CAREER), CNS-1218475 and CNS-1218154, in part by ONR grant N00014-11-1-0898, and in part by Singapore's Agency for Science, Technology and Research (A⋆STAR) under the Human Sixth Sense Programme.
Authors' addresses: J. Chen, Y. Wang, G. Xing, and B. Punch, 428 S. Shaw Ln., RM 3115, East Lansing, MI 48824-1226, USA; emails: {chenjinz, wangyu3}@msu.edu, {glxing, punch}@cse.msu.edu; R. Tan, Advanced Digital Sciences Center, 1 Fusionopolis Way, #08-10 Connexis North Tower, Singapore 138632; email: tanrui@adsc.com.sg; X. Wang, Department of Electrical and Computer Engineering, The Ohio State University, 508 Dreese Laboratories, 2015 Neil Avenue, Columbus, OH 43210, USA; email: xwang@ece.osu.edu; X. Wang, Department of Electrical and Computer Engineering, The Ohio State University, 805 Dreese Laboratories, 2015 Neil Avenue, Columbus, OH 43210, USA; email: wang.3570@osu.edu; D. Colbry, Institute for Cyber-Enabled Research, Michigan State University, Biomedical and Physical Sciences Building (BPS), 567 Wilson Road, Room 1440, East Lansing, MI 48824-1226, USA; email: colbrydi@msu.edu.

DOI: http://dx.doi.org/10.1145/2675353

ACM Reference Format:
Jinzhu Chen, Rui Tan, Yu Wang, Guoliang Xing, Xiaorui Wang, Xiaodong Wang, Bill Punch, and Dirk Colbry. 2014. A sensor system for high-fidelity temperature distribution forecasting in data centers. ACM Trans. Sensor Netw. 11, 2, Article 30 (December 2014), 25 pages.
DOI: http://dx.doi.org/10.1145/2675353

1. INTRODUCTION

Data centers have become a critical computing infrastructure in the era of cloud computing. Research has shown that more than 23% of data center outages are caused by servers' self-protective shutdowns because of overheating [Aperture Research Institute 2007]. For instance, Wikipedia, a popular online encyclopedia, went down on March 24, 2010 because of server overheating [WikiMedia Foundation 2010]. Currently, the common practice to prevent overheating is to overcool server rooms. Due to such a conservative strategy, the cooling systems consume excessive power, which makes up to 50% of the total energy consumption in many data centers [U.S. Environmental Protection Agency 2007].

Various thermal management schemes for improving the energy efficiency of data centers rely on real-time and high-fidelity ambient temperature monitoring [Bash et al. 2006; Bash and Forman 2007; Moore et al. 2005; Tang et al. 2008]. However, the existing data centers typically have limited temperature monitoring capability. For instance, the servers in many legacy data centers are equipped with motherboard temperature sensors only, which cannot accurately measure the ambient temperature [Liang et al. 2009]. Recently, the Wireless Sensor Network (WSN) has been identified as an ideal enabling technology for thermal monitoring in data centers due to several of its salient advantages, including sufficient coverage and no reliance on additional network and facility infrastructure in already complicated data center environments. However, precise temperature monitoring alone may not be sufficient for preventing unexpected server shutdowns because various thermal emergencies may quickly cause overheating. Therefore, it is important to design temperature prediction systems to forecast potential overheating events such that the thermal actuators (e.g., the cooling systems) have enough time to react. Moreover, the prediction system can also send alarm messages to the data center administrators for human intervention if necessary. In addition to overheating alarms, proactive thermal control systems for data centers [Chen et al. 2013] can be built on real-time temperature prediction, through which energy efficiency and hardware reliability can be significantly improved. Specifically, with accurately predicted temperature evolution in the near future, cooling systems can safely increase the temperature setpoints without leading to server overheating and thus improve energy efficiency in data centers [Bash et al. 2006]. Moreover, proactive control can reduce transient temperature variation, which is shown to contribute to the hardware failure rates in data centers [El-Sayed et al. 2012].

However, several major challenges must be addressed in designing a real-time and high-fidelity temperature prediction system. First, data centers are complex Cyber-Physical Systems (CPS) whose thermal characteristics are inherently affected by both physical (e.g., complex airflows and server deployment layout) and cyber (dynamic server workloads) factors. Therefore, prediction algorithms based on simplified physical and cyber models would not yield satisfactory performance. Second, the number of locations where temperatures are of particular interest (e.g., the inlets and outlets of all servers) is often large, making it prohibitively expensive to deploy a sensor at each such location. It is challenging to reduce the number of sensors while maintaining satisfactory temperature prediction accuracy. Third, it is desirable to decouple the prediction system and the computing resources of the monitored data center. This design not only avoids potential interruptions to the prediction system due to unexpected

server shutdowns, but also improves system portability. To this end, the prediction system must operate on limited computing resources while maintaining high prediction fidelity.

To address these challenges, we propose a novel cyber-physical approach that integrates *in situ* wireless sensors, transient Computational Fluid Dynamics (CFD) modeling [Wendt 1995], and real-time data-driven prediction algorithms. CFD is a widely adopted numerical tool that can simulate the future evolution of temperature distribution in data centers. However, without accounting for runtime behaviors of the data center, CFD has highly variable accuracy, poor scalability, and prohibitive computational complexity, which make it ill-suited for high-fidelity online prediction. To overcome these limitations, our approach leverages the realistic physical thermodynamic models provided by CFD to generate simulated temperature distribution, which is then integrated with the real sensor measurements to train the real-time prediction algorithm. Moreover, unlike traditional thermal management solutions where CFD is used in an open-loop fashion, our approach utilizes real sensor feedback to calibrate the CFD simulation results. Our approach has the following advantages.

First, by leveraging transient CFD modeling, our approach ensures the fidelity of predicting many rare but critical thermal emergencies (e.g., cooling system failures) that may not be captured by real sensors in operational data centers. Second, by integrating realistic physical CFD models and real sensor measurements, our approach only requires prediction algorithms with low computational complexity. This enables the development of portable thermal management systems that do not rely on the infrastructure of the monitored data center. Finally, because CFD can simulate the temperature at uninstrumented locations, our approach significantly reduces the number of sensors without leading to substantial degradation of prediction fidelity.

We implemented our temperature prediction system using 36 wireless motes equipped with temperature and airflow sensors. We deployed our system in a single-rack testbed composed of 15 running servers and in a small-scale production data center testbed composed of five racks and 229 servers. The extensive evaluation shows that our system can predict the temperature evolution of servers with highly dynamic workloads at an average error of 0.52°C, within a duration up to 10 minutes.

2. RELATED WORK

Existing approaches for data center thermal management can be broadly divided into two groups. The first group of approaches focuses on assignment of server workload based on physical thermodynamic models to improve the energy efficiency of data centers. In Bash and Forman [2007], cooling-efficient servers are identified and assigned with more workload. In Moore et al. [2005], heat recirculation is minimized by distributing computing power to those servers with less heat recirculation at their inlets. Tang et al. propose abstract models to distribute computing power in a data center by minimizing peak inlet temperatures [Tang et al. 2008].

The second group of approaches focuses on the modeling and prediction of temperature distribution to prevent thermal emergencies. In Heath et al. [2006] and Ramos and Bianchini [2008], the temperature distribution of a single server is emulated based on simplified thermodynamic laws, CPU temperature/utilization, and airflow velocity. In Tang et al. [2006], a heat flow model is proposed to characterize heat recirculation and predict the temperature distribution. In Moore et al. [2006], an artificial neural network is employed to learn and predict steady-state temperature distribution under static workload assignment. However, these approaches rely on steady-state thermal models that cannot well model temperature evolution when the heat dissipation from servers is dynamic. They also require a controlled training procedure that is usually intrusive or even infeasible in a production data center. Moreover, such data-driven

approaches often suffer low prediction fidelity due to insufficient training data, especially for rare but critical thermal emergency conditions like cooling system failures. In Choi et al. [2007], a CFD model is constructed to enable offline temperature prediction and analysis for critical thermal emergencies but without online prediction capability. In Li et al. [2011], a forecasting model, called ThermoCast, predicts the temperature distribution in the near future based on a simplified thermodynamic model. However, the model relies on several specific assumptions on airflow dynamics that may not hold in diverse data center environments. For instance, it assumes that the cold air runs vertically from raised floor tiles. This does not hold in many data centers where the cooling equipment is placed in the rows of the racks [Niemann 2006; Bell 2012] or near the racks [Rasmussen 2011], which generates significant side-to-side airflow. A recent work [Jonas et al. 2012] predicts the temperature as a weighted average of contributing temperatures of each heat source, where the model coefficients are determined via offline CFD simulations. Thus, this approach does not leverage the real temperature measurements in a data center to ensure prediction accuracy.

Several sensor systems have been developed for temperature monitoring in data centers [Liang et al. 2009; Choochaisri et al. 2010]. RACNet [Liang et al. 2009] is designed for reliable data collection in large-scale data centers, where each node is connected with multiple daisy-chained temperature sensors. In Wang et al. [2011], a fusion-based approach is developed to detect hot spots in data centers using measurements of multiple sensors. Robotic systems have also been designed to roam inside the data centers for plotting thermal maps [Mansley et al. 2011] and energy management [Lenchner et al. 2011]. In Biswas et al. [2011], thermoelectric coolers on server processors are used to remove heat directly from hot processors. However, these studies are not concerned with real-time temperature prediction or could not be used for legacy data centers where the servers are not equipped with thermoelectric coolers.

3. PROBLEM STATEMENT AND APPROACH OVERVIEW

3.1. Problem Statement

The temperatures at the inlet and outlet of a server are critical thermal conditions for the operation of the server. The inlet temperature is often defined as the server's operating ambient temperature, which must be within a small range (e.g., 15°C to 27°C [ASHRAE Technical Committee 9.9 2011]). The outlet temperature characterizes the amount of heat that needs to be removed by Air Conditioners (ACs) to avoid overheating. Therefore, in this work, we aim to predict the temperatures at the inlets and outlets of the servers of interest. The set of these temperatures is referred to as *temperature distribution*. The accurate prediction of the temporal evolution of temperature distribution is challenging because of the complex thermal and air dynamics in data centers. Specifically, the dynamic workload and other server activities (e.g., disk and network access) generate different amounts of heat over time. The heat is dissipated by extremely complex airflows that are driven by server internal fans and ACs. Moreover, the heat dissipation is highly dependent on the racks and other physical structures in a data center.

Our temperature distribution forecasting system is designed to meet the following objectives. (1) ***High fidelity***. We aim to achieve high prediction fidelity with about a 1°C error bound. This requirement ensures that the predicted temperature will not trigger excessive false overheating alarms or miss real overheating events. Moreover, as shown in Moore et al. [2005], a 1°C increase of the maximum server inlet temperature can lead to 10% higher cooling costs. Therefore, high prediction fidelity allows servers to operate with less conservative temperature setpoints, improving the energy efficiency of data centers. (2) ***Long prediction horizon***. The system should achieve satisfactory

prediction accuracy over a considerably long time duration (e.g., 10 minutes), referred to as the *prediction horizon*, into the future. We focus on providing a prediction horizon in the order of minutes. This is motivated by the fact that it usually takes up to several minutes to reach the overheating temperature [Active Power, Inc. 2007] in thermal emergencies (e.g., excessive server overload or AC failure). This provides enough time for the thermal actuators (e.g., ACs) to prevent overheating, as well as for data center administrators to make necessary interventions. However, a longer prediction horizon often requires the sacrifice of prediction fidelity. (3) ***Full coverage of thermal conditions***. Our approach is designed to predict the temperature distribution under normal working conditions of the data center, in which overheating is mainly due to high workload, as well as those abnormal and emergency situations (e.g., AC failures) that can lead to catastrophic consequences. (4) ***Timeliness and low overhead***. To enable prompt actuation, the prediction should be performed in an online fashion with tight real-time requirements. The overhead of the prediction system should be affordable for low-end servers, desktop computers, or even embedded computing devices so that the system can be easily deployed and operated in a noninvasive fashion and without relying on the infrastructure of the monitored data center.

CFD [Wendt 1995] is a widely used tool to guide data center layout and cooling system design. The predictive nature of CFD also allows the user to simulate the future evolution of temperature distribution. However, CFD has the following two major limitations. First, the accuracy of CFD highly depends on how well the adopted thermodynamic models reflect the realities. Considerable expertise and labor-intensive fine- tuning are often required in the modeling process, which makes fine-grained CFD simulation intractable for medium- to large-scale data centers. Moreover, CFD often has considerable temperature modeling errors that range from 2°C to 5°C [Wang et al. 2011; Singh et al. 2010]. This often leads to highly conservative temperature setpoints, resulting in excessively high power consumption by cooling systems in a data center. Second, CFD has high computational complexity that prohibits it from temperature prediction at runtime. For instance, it can take 5 minutes on a high-end 12-core server to simulate 5 seconds of the temperature evolution of a rack equipped with 15 servers. As a result, CFD alone is not sufficient for high-fidelity and real-time temperature prediction in data centers.

3.2. Approach Overview

Our approach integrates *in situ* wireless/on-board sensors, transient CFD simulation, and real-time prediction modeling to achieve high-fidelity temperature prediction in data centers. The sensors collect environment temperature and airflow velocity data at various physical locations (e.g., server inlets, outlets, fans, raised floor tiles, AC cold air inlets and hot air outlets). The collected data are then used to train the time series prediction models. To ensure modeling fidelity, multiple models are used to capture normal and various abnormal working states, such as the failure of different AC units. A challenge for such a training-based approach is to collect sufficient datasets that cover various thermal conditions, especially for those abnormal and emergency situations that rarely happen but have catastrophe consequences. The controlled experiments for generating these situations are often intrusive or even harmful in operational data centers. To address this issue, we leverage transient CFD simulation, which is capable of simulating any overheating condition, to generate additional training data for the prediction models. This approach avoids running the computationally intensive CFD in an online fashion, yet preserves the realistic physical characteristics of the training data. The CFD simulation results are also calibrated by runtime sensor measurements. As a result, our approach only requires moderately accurate CFD modeling, thus significantly reducing the efforts of CFD model tuning. Another advantage of our approach

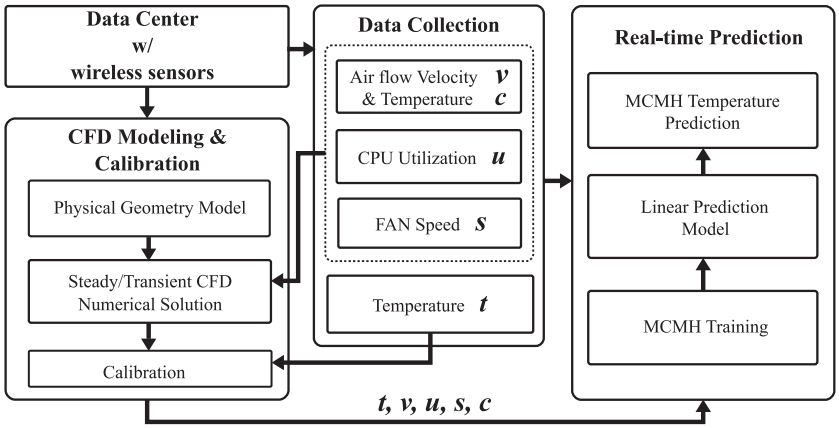

Fig. 1. Prediction system architecture.

is that, by integrating CFD simulation and runtime sensor measurements, the number of required sensors is significantly reduced, leading to lower deployment costs and less intrusiveness to production data centers.

Figure 1 illustrates the architecture of our prediction system. The system consists of three major components. (1) ***Data collection***. This component periodically collects the measurements of CPU utilization and server fan speed through on-board sensors while using a WSN to collect the measurements of temperatures and airflow velocities. The historical measurements are used to calibrate the CFD modeling and train the prediction models, whereas the runtime sensor measurements are fed to the real-time prediction component to predict temperatures. (2) ***CFD modeling and calibration***. In addition to sensor measurements, a key feature of our system is to leverage the transient CFD simulation to compute fine-grained temperature evolution, which assists the training of the time series prediction models. Our system uses *in situ* sensor measurements to calibrate the transient CFD simulations and generate calibrated temperature time series data for normal and various abnormal thermal conditions, such AC failures. These results are then fed as training data to the real-time prediction component. (3) ***Real-time multichannel and multihorizon temperature prediction***. The real-time prediction component constructs time series prediction models with training data from both historical measurements and CFD simulations, and it outputs the runtime temperature predictions. Although complex nonlinear models may achieve good prediction accuracy, they often have high complexity. Our solution uses multiple simple linear models to approximate the complex nonlinear thermodynamic laws. For each different major thermal condition (hereafter referred to as a *channel*), such as the failure of different AC units, multiple prediction models with different prediction horizons are constructed. Different prediction horizons in our system give administrators more flexibility in implementing thermal actuators (e.g., taking appropriate measures in an incremental fashion).

4. CFD MODELING AND CALIBRATION

In this section, we first briefly introduce CFD and then present a case study of modeling a rack of servers using CFD. The case study helps us understand the major limitations of CFD. We then present an approach to calibrating CFD using real sensor measurements.

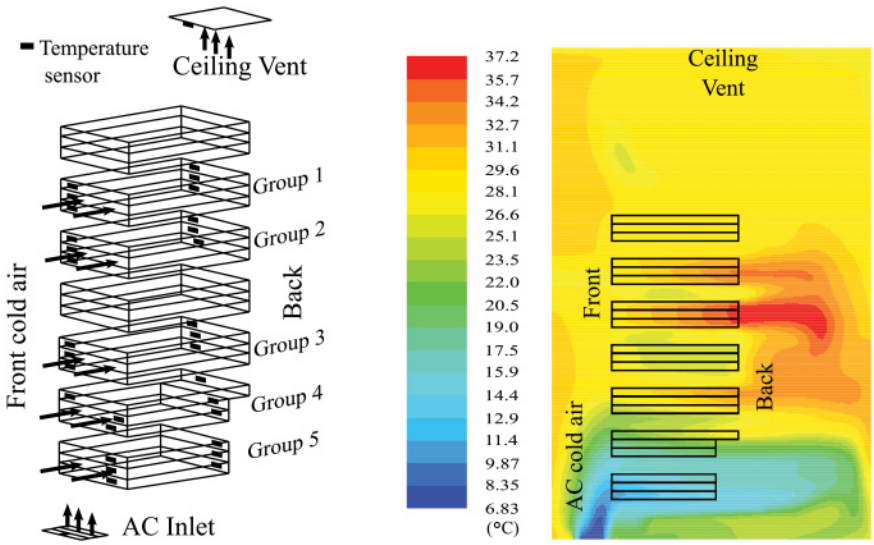

Fig. 2. (a) Server geometries with temperature sensor locations; (b) Side view of the steady-state temperature map when the servers in Group 1 and Group 2 are running with full utilization.

4.1. Background on CFD

CFD is a widely adopted numerical tool to simulate future temperature evolution. It iteratively solves a system of fluid and heat transfer equations in the form of nonlinear partial differential equations under the constraints of mass, momentum, and energy conservation. The nonlinear nature of these equations and the complex boundary conditions in data center environments (e.g., the physical structures) usually make it impossible to solve these equations analytically. Therefore, CFD typically solves these equations using numerical approaches. Specifically, by dividing the continuous fluid field into small cells, CFD solves the fluid and heat transfer equations in each cell with significantly simplified boundary conditions. The global optimal solution is found iteratively, in which all cells meet the convergence requirements, thus giving the steady-state temperature distribution. For the transient simulation, the model is also discretized into small time steps in the time domain. At each time step, CFD iteratively finds the global optimal solution, giving the transient temperature distribution. The boundary conditions such as AC airflow temperature, velocity, and power consumption of servers can also be set for each time step with user-defined values (e.g., sensor measurements) so that CFD can simulate any normal or abnormal thermal situations. Therefore, the accuracy of transient CFD modeling is particularly important for achieving high prediction fidelity.

4.2. A Case Study

We now present a case study using CFD to model a testbed of rack servers, which helps us understand the performance limitations of CFD. Figure 2(a) shows the physical geometry of a rack server testbed. A total of 15 servers on the rack are grouped into five server groups. The detailed settings of the server rack can be found in Section 6. For CFD modeling, we use wireless sensors to measure the boundary conditions, including the temperatures and velocities of the air discharged by the AC and exhausted by the ceiling vent. A total of 30 sensors are deployed to measure the temperature distribution

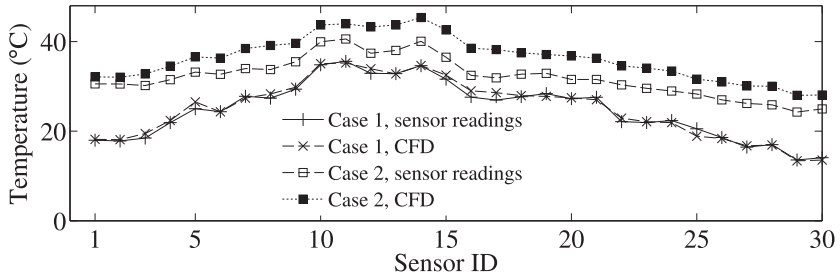

Fig. 3. Real sensor readings and CFD prediction. Case 1: Servers in Group 1 and Group 2 run with full utilization; Case 2: AC failure.

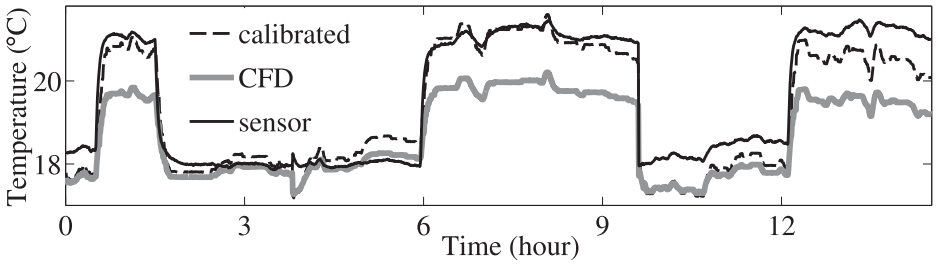

Fig. 4. Transient temperatures at the outlet of the lowest server (*sensor*: real sensor readings; *CFD*: transient simulation result of CFD; *calibrated*: calibrated transient simulation result of CFD).

around the rack. Nodes 1–15 are installed at the outlets of the servers, and Nodes 16–30 are installed at the inlets of the servers.

Figure 2(b) shows the steady-state temperature map calculated by CFD software (Fluent) when servers in Group 1 and Group 2 are running with full utilization (referred to as Case 1). We can see that the cold air is mostly drawn by the lower servers, and the two groups of servers running with full utilization have much higher exhaust air temperatures than other servers. Figure 3 plots the sensor readings as well as the temperatures calculated by CFD. We can see that, for Case 1, CFD can accurately predict the steady-state temperature distribution. The Root-Mean-Square Error (RMSE) across all sensors is only 0.7°C. The result in Figure 3 is achieved by extensively tuning CFD with the help from an expert with 20 years of experience in CFD modeling. For instance, the exhaust airflow of servers, the cooling airflow of the AC, and the corresponding sensor locations in the CFD physical model were carefully adjusted in a number of iterations. We note that such an extensive tuning process is a common practice for constructing CFD for real data centers. We then use the well-tuned CFD to predict the steady-state temperature distribution for the case of AC failure (referred to as Case 2). Figure 3 shows that the CFD exhibits considerable errors (RMSE of 4.4°C) in case of AC failure. In addition to the steady-state prediction, we also examine the accuracy of CFD in a transient simulation, which is critical for the performance of real-time prediction. Figure 4 shows the temporal evolution at the location of sensor 1 computed by CFD, as well as the real readings from sensor 1. During this period, the CPU utilizations of servers are varied, resulting in highly dynamic temperatures at this sensor location. It can be clearly seen that the CFD result contains significant biases with respect to the real sensor readings. The major reason for those errors is that CFD does not exactly model the true data center environment and all its important system parameters (e.g., material properties). In practice, it is extremely difficult and labor-intensive to construct a CFD model that is accurate in all thermal

conditions. Therefore, to make CFD practical in our prediction system, we discuss in Section 4.3 how to calibrate the temperature data simulated by CFD using real sensor measurements collected in the data center. Such calibration significantly reduces the dependency of prediction performance on CFD modeling.

4.3. CFD Calibration

The results from Section 4.2 show that the CFD simulation exhibits considerable errors, particularly in transient-state simulations. To address this limitation, we propose calibrating the CFD simulation results using runtime sensor measurements. By denoting x_i and y_i as the temperature calculated by CFD and the calibrated temperature at the position of sensor i, the calibration function is given by $y_i = \sum_{k=0}^{K} a_{i,k} \cdot x_i^k$, where K is the order of the calibration function and $a_{i,k}$ are the coefficients to be learned from training data. By providing real sensor data as y_i, the coefficients $a_{i,k}$ can be learned based on the least-square criterion. For each sensor, a calibration function is constructed as long as there are sufficient real sensor measurements collected. As an example, we use the first 3 hours of data in Figure 4 to construct the calibration function for each sensor and then use all the data for testing. Figure 4 also shows an example of calibrated CFD results with $K = 1$.

5. REAL-TIME TEMPERATURE PREDICTION

This section first presents our approach of predicting temperature distributions using a linear prediction model and then discusses the training of the prediction model.

5.1. Real-Time Prediction Model

Suppose that wireless temperature sensors are deployed at the inlets and outlets of a total of N monitored servers. The temperature distribution is defined as $\mathbf{t} = [t_{\text{in}}^1; t_{\text{out}}^1; \ldots; t_{\text{in}}^N; t_{\text{out}}^N] \in \mathbb{R}^{2N \times 1}$, where t_{in}^n and t_{out}^n denote the temperatures at the inlet and outlet of the n^{th} server. The prediction model should include the observable variables that significantly affect $\mathbf{t}$ to achieve the accurate prediction of $\mathbf{t}$. In this work, our prediction model accounts for the temperatures (denoted by $\mathbf{c}$) and velocities (denoted by $\mathbf{v}$) of the cold airflow distributed by the ACs, CPU utilization (denoted by $\mathbf{u}$), and internal fan speeds (denoted by $\mathbf{s}$) of all monitored servers. Moreover, the historical temperature distributions also largely affect the temperature distributions in the near future. Therefore, we define the *state* of the monitored servers at a time instance, denoted by $\mathbf{p}$, as the concatenation of $\mathbf{t}$, $\mathbf{c}$, $\mathbf{v}$, $\mathbf{u}$, and $\mathbf{s}$. Specifically, $\mathbf{p} = [\mathbf{t}; \mathbf{c}; \mathbf{v}; \mathbf{u}; \mathbf{s}]$. Our approach can be easily extended to include other observable variables to address various kinds of servers. For instance, hard disc access rates can play an important role in the temperature distribution of file servers.

We assume that each variable in $\mathbf{p}$ can be measured periodically and synchronously by multiple sensors. In the rest of this article, the period of data collection is referred to as the *time step*. Intuitively, the most recent states significantly affect the current and the future states. In our approach, we predict the temperature distribution at time step $(t + k)$ based on the most recent R states, where $t \in \mathbb{Z}$ denotes current time step, and $k \in \mathbb{Z}$ is referred to as the *prediction horizon*. For a given k, we assume that the predicted temperature distribution[1] at time step $(t + k)$ is given by $\hat{\mathbf{t}}(t + k) = f_k(\mathbf{p}(t), \mathbf{p}(t-1), \ldots, \mathbf{p}(t-R+1))$, where $f_k(\cdot)$ is the function characterizing the physical law governing the thermodynamic process. However, $f_k(\cdot)$ is often difficult to find in practice due to the highly complex data center environment. In this work, we propose a linear prediction model to approximate $f_k(\cdot)$, which allows the online

[1]For clarity of presentation, we let $\hat{x}$ denote the *predicted* value of x.

real-time prediction at low overhead. Supposing $\mathbf{p} = [p_1; p_2; \ldots]$, define $\mathbf{p}^s = [p_1^s; p_2^s; \ldots]$ where $s \in \mathbb{Z}$. Moreover, we define $\mathbf{q}(t) = [\mathbf{p}(t); \mathbf{p}^2(t); \ldots; \mathbf{p}^s(t)]$ and $\mathbf{x}(t) = [\mathbf{q}(t); \mathbf{q}(t-1); \ldots; \mathbf{q}(t-R+1)]$. According to Taylor's theorem, the high-order Taylor polynomial can well approximate a function. The s^{th} order Taylor polynomial of $f_k(\cdot)$ is given by the linear combination of all the combinatorial terms of the elements in $\mathbf{x}(t)$, which, however, results in exponential complexity with respect to N. Therefore, we ignore all cross terms in the Taylor polynomial and adopt the following linear prediction model:

$$\hat{\mathbf{t}}(t+k) = \mathbf{A}_k \cdot \mathbf{x}(t), \tag{1}$$

where $\mathbf{A}_k \in \mathbb{R}^{2N \times M}$, and M is the length of $\mathbf{x}(t)$.

Since only the arithmetic calculations are involved in Equation (1), the prediction can be efficiently computed even on low-power embedded platforms. Note that $\mathbf{A}_k$ is different for each prediction horizon k. By setting increasing prediction horizons, Equation (1) predicts the temporal evolution of the temperature distribution. Intuitively, because the correlation between $\mathbf{t}$ and $\mathbf{p}$ decreases over time in a dynamic environment, the prediction with a larger k becomes less accurate.

5.2. Model Training

During the normal running state of the data center, the training data are collected from the wireless sensors (e.g., temperature and airflow velocity) or server on-board sensors (e.g., CPU utilization and fan speed). In addition to sensor data, CFD data are generated for normal and abnormal running states by manually giving different boundary conditions to the CFD transient simulations. For example, different ACs can be shut down during the CFD transient simulation. Suppose a dataset with a time step index from 1 to L is collected after system deployment or generated by CFD to train the linear model $\mathbf{A}_k$ for any given k. We adopt the least-square criterion to train $\mathbf{A}_k$. Specifically,

$$\begin{aligned}\mathbf{A}_k &= \arg\min_{\mathbf{A}_k} \sum_{t=R}^{L-k} \|\mathbf{t}(t+k) - \hat{\mathbf{t}}(t+k)\|_{\ell_2}^2 \\ &= \arg\min_{\mathbf{A}_k} \sum_{t=R}^{L-k} \|\mathbf{t}(t+k) - \mathbf{A}_k \cdot \mathbf{x}(t)\|_{\ell_2}^2,\end{aligned}$$

where $\|\cdot\|_{\ell_2}$ represents the Euclidean norm. A desirable property of this formulation is that the problem can be decomposed to the subproblems of finding the rows of $\mathbf{A}_k$ separately. The separation can significantly reduce the computation complexity in training. By denoting $\mathbf{a}_j$ as the j^{th} row of $\mathbf{A}_k$ and $t_j(t+k)$ as the j^{th} element in $\mathbf{t}(t+k)$, the subproblem is

$$\mathbf{a}_j = \arg\min_{\mathbf{a}_j} \sum_{t=R}^{L-k} \left(t_j(t+k) - \mathbf{a}_j \cdot \mathbf{x}(t)\right)^2. \tag{2}$$

The closed-form solution of $\mathbf{a}_j$ is $\mathbf{a}_j = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{t}_j$, where $\mathbf{X} = [\mathbf{x}(R), \mathbf{x}(R+1), \ldots, \mathbf{x}(L-k)]^\top$ and $\mathbf{t}_j = [t_j(R+k); t_j(R+k+1); \ldots; t_j(L)]$. The matrix $\mathbf{A}_k$ can be constructed once all its rows are computed.

We now discuss two practical issues related to model training.

5.2.1. Regularized Regression. Because the variables in the state $\mathbf{p}$ may be affected by the same thermal conditions, they may be correlated with each other. This is called *multicollinearity* in regression analysis. Multicollinearity can lead to inflation of the

estimated coefficients in $\mathbf{a}_j$ and hence adversely affects the performance of model training. A common approach to deal with multicollinearity problem is to use regularized regression [Hoerl and Kennard 1970]. Specifically,

$$\mathbf{a}_j = \arg\min_{\mathbf{a}_j} \sum_{t=R}^{L-k} (t_j(t+k) - \mathbf{a}_j \cdot \mathbf{x}(t))^2 + \lambda \|\mathbf{a}_j\|^2,$$

where λ is the *regulation factor*. By including the norm of $\mathbf{a}_j$ in the minimization, the inflation of the coefficients in $\mathbf{a}_j$ can be effectively restricted. The closed-form solution of $\mathbf{a}_j$ is $\mathbf{a}_j = (\mathbf{X}^\top \mathbf{X} - \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{t}_j$. As shown in Section 7.1.3, with proper settings for λ, the regulation can improve the performance of the model training.

5.2.2. Training Data Generation Using CFD. A practical issue about training data generation using CFD is how to generate sufficient training data to ensure that the trained model well captures the underlying thermal dynamics and thus delivers accurate predictions. A naive solution is to generate training data to fully cover all possible thermal conditions. For instance, to generate training data for the channel that addresses AC failure, the naive solution needs to simulate the AC failure under all possible initial states. However, a major challenge here is that the state $\mathbf{p}$ has combinatorial complexity. Due to the high dimensionality of $\mathbf{p}$, enumerating all possible initial states in the generated training data will incur extremely high computation overhead. Intuitively, a certain amount of simulated data traces with initial states that sparsely span in the state space may be sufficient to train the linear regression model. Based on this intuition, we generate data traces with a random initial state $\mathbf{p}$ that is uniformly distributed within its possible range. The number of generated data traces should be large enough to prevent overfitting. In Section 7.1.6, our experiments show that 10 transient data traces generated by CFD are sufficient for training the model characterizing AC failure for a rack of 15 servers.

5.3. Dimension Reduction

A monitored temperature instance (i.e., an element of the temperature distribution $\mathbf{t}$) may not be strongly correlated with every other variable in the state $\mathbf{p}$. For small-scale deployments such as a rack, the dimension of the state $\mathbf{p}$ is limited. With sufficient training data, the regression result can accurately characterize the correlations between the monitored temperature and every other variable in $\mathbf{p}$. However, for large-scale deployments, such as a server room with many racks, the dimension of $\mathbf{p}$ is high. For instance, on our small-scale production data center testbed (cf. Section 7.2), the dimension of $\mathbf{p}$ is 229. As a result, the regression is prone to overfitting, which leads to poor prediction performance. Therefore, before constructing the prediction model in Equation (1), it is desirable to choose a subset of variables in $\mathbf{p}$ that are most correlated to the monitored temperature to increase the prediction accuracy. In this work, we present a dimension reduction approach that adopts the *partial correlation* metric [Stuart et al. 2009] to rank the variables in $\mathbf{p}$ regarding their correlation with the monitored temperature. We note that other approaches may also be applicable to the dimension reduction problem [Van der Maaten et al. 2009]. Our approach performs dimension reduction based on the variable ranking for each monitored temperature separately. Specifically, let n denote the dimension of state $\mathbf{p}$ and p_i denote the i^{th} element of $\mathbf{p}$. To compute the partial correlation of the monitored temperature t_j and a variable p_i, we first construct the linear regression models for predicting t_j and p_i based on the remaining variables in $\mathbf{p}$ (i.e., $\{p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n\}$) using the approach described in Section 5.2. Let $\hat{t}_j$ and $\hat{p}_i$ denote the predictions based on the remaining variables in $\mathbf{p}$. The partial correlation, denoted by $\rho(t_j, p_i)$, is given by the correlation

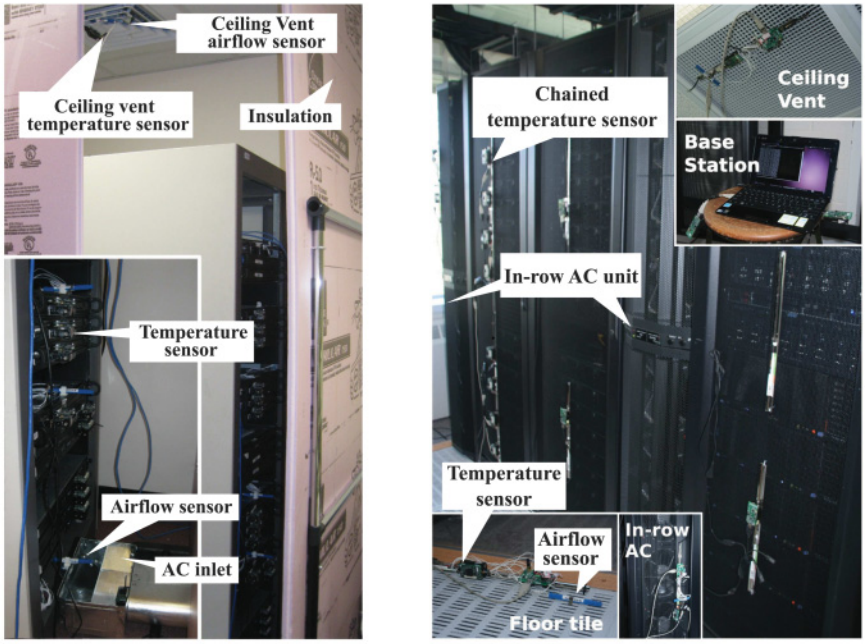

Fig. 5. Testbeds. (a) Single-rack testbed; (b) Production testbed (HPCC).

coefficient of the errors in predicting t_j and p_i using remaining variables; that is,

$$\rho(t_j, p_i) = \mathrm{r}(t_j - \hat{t}_j, p_i - \hat{p}_i),$$

where r(·) measures the correlation coefficient. Therefore, the $\rho(t_j, p_i)$ measures the partial correlation between t_j and p_i at a particular horizon while the effects from all other input variables are removed. After calculating the partial correlations $\{\rho(t_j, p_i)|i \in [1, n]\}$, we choose a subset of variables in $\mathbf{p}$ with the *highest* partial determination coefficients (i.e., $\rho^2(t_j, p_i)$). With the chosen variables, we construct the linear regression model in Equation (2), where the elements of $\mathbf{a}_j$ corresponding to the unselected variables are set to zero. Note that the dimension reduction result varies with prediction horizon k.

6. SYSTEM IMPLEMENTATION AND DEPLOYMENT

We implemented the proposed system and deployed it on two testbeds. Here, we first describe the set-up of the two testbeds and then discuss the system implementation.

6.1. Testbeds and Sensor Deployment

Our first single-rack testbed, shown in Figure 5(a), consists of a rack of 15 1U[2] servers in a 5 × 6-square-foot room insulated by foam boards. Two types of servers (4 Dell PowerEdge 850 nodes and 11 Western Scientific nodes) are placed on the rack. The rack is placed directly under an infrastructure ceiling vent that exhausts the hot air out of the room. A portable AC made by Tripp Lite, Inc. (model SRCOOL12K) is placed outside the room. It delivers cold air through the AC inlet located at the bottom of the room in front of the rack, which is consistent with the cooling airflow used in

[2]U is the unit of the height of a server, which is 1.75 inches.

the popular raised-floor cooling design. On the rack, the 15 servers are grouped every three servers with a 2U distance between every adjacent two groups. A total of 15 Iris [Memsic Corp. 2012] temperature sensors are mounted with brackets at the inlets of the group of five servers, and another 15 temperature sensors (eight Iris and seven TelosB [Memsic Corp. 2012]) are mounted with brackets at the outlets of these servers. At the ceiling vent, a temperature sensor (TelosB) is mounted with a bracket, and a F333 airflow velocity sensor [Degree Controls, Inc. 2011] is taped to face the exhausting airflow. To monitor the AC cold airflow, we place a temperature sensor (Iris) in the AC inlet register and tape an identical airflow velocity sensor in front of the register. This small testbed allows us to study the fine-grained thermal dynamics of a single rack. Moreover, by controlling the AC system, the testbed can emulate various thermal emergency scenarios.

Figure 5(b) shows the second testbed in a server room of High Performance Computer Center (HPCC) at Michigan State University. The testbed consists of 229 servers with 2,016 CPU cores on five server racks. Those racks are arranged in two rows with a cold aisle between them. One row of racks is shown in Figure 5(b). In addition to the raised-floor cooling system that blows cold air vertically from the floor tile into the cold aisle, two in-row AC cooling units are installed between the racks for each row, which produce major cold air at different heights and generate significant side-to-side airflow. To prevent major hot air recirculation, two pieces of glass wall are installed at the end of the cold aisle. We chain the sensors and mount them at both the front and rear doors of the server racks to monitor the inlet and outlet temperatures, respectively. For one rack, we evenly deploy eight sensors to monitor the server inlets and eight sensors to monitor the server outlets. For other racks, we mount one or two sensors to monitor the server inlets and outlets at different heights. We monitor two out of four in-row AC units by mounting a bundle of temperature sensors and airflow sensors at cold air inlets. Another two bundles are fixed at the floor tile and the ceiling vent. The details of sensor deployment can be found in Figure 16.

6.2. Implementation of the Sensor Network

Wireless Sensors: In each of our testbed implementations, we use a single-hop network architecture in which the base station sends data collection requests to sensors sequentially, and each sensor transmits the measurements. Every 5 seconds, the base station performs a round of sequential data collection from all sensors. We note that a multihop network topology can be employed when more server racks need to be monitored. Because this collection scheme works in a time-division fashion, the system does not generate many collisions between the data transmissions of different sensors. TelosB [Memsic Corp. 2012] and Iris [Memsic Corp. 2012] motes are used for collecting temperature data. To collect the airflow velocity data, we connect the Senshoc mote, an implementation of the open design of TelosB, to a standalone air velocity sensor [Degree Controls, Inc. 2011] via an I^2C interface. The programs on these motes are implemented in TinyOS 2.1 [Levis et al. 2005].

On-board Sensors: CPU utilization and fan speed are two important thermal variables that the system needs to collect from the on-board sensors of each server. Data centers typically run various server-monitoring utility tools (e.g., `atop`, `ganglia`) that can collect on-board sensor information. These tools are used to implement the data collection of CPU utilization and fan speed for our production testbed. In our single-rack testbed, we implement a simple program to control and measure CPU utilization and report fan speed from `lm-sensors` utilities, which are commonly available in `GNU/Linux` distributions. Similar to the wireless sensor data collection, the base station requests the CPU utilization and fan speed from each server sequentially. However, instead of

using wireless links, the base station takes advantage of the existing Ethernet infrastructure to collect these on-board sensor data.

6.3. Discussion

We now discuss the costs and benefits of deploying our temperature prediction system in data centers. Our prediction system comprises low-cost wireless sensor nodes and PC-class base stations. Moreover, it is even more cost-effective in newer data centers where the servers may have already been equipped with inlet temperature sensors. In addition to the costs of the WSN, our system requires computer-intensive CFD simulations to assist in temperature prediction, where the major costs are due to the computing resources, labor of model construction, and CFD software license. Since our system only requires offline CFD simulations, it can leverage the existing data center computing resources during off-peak hours to train the prediction models. Once the models are constructed, our system will no longer consume any computing resources in the data center. In addition, many data centers have already incorporated CFD analysis in their thermal design, which can be reused to reduce the extra labor and license costs for CFD.

The benefits of deploying our prediction system are twofold. First, heat-induced server shutdown contributes to more than 23% of server outages in data centers [Aperture Research Institute 2007]. It equals US$ 0.2 million in loss per year for a data center on average [Emerson Network Power 2011]. With accurate multihorizon temperature prediction, data center administrators can be alerted to potential thermal emergencies (e.g., server overheating), thus allowing more time for necessary actions such as migrating server workload to prevent these emergencies. Second, real-time temperature prediction could enable proactive thermal control at a higher room temperature in data centers without causing server overheating, thereby achieving as much as 30% of total energy saving [Chen et al. 2013]. For instance, the proactive thermal control system can dynamically adjust the cold air temperatures and velocities of the cooling system based on predicted future temperatures instead of on measured temperatures.

7. PERFORMANCE EVALUATION

To evaluate the performance of our prediction system, we conduct extensive experiments on the single-rack testbed and the small-scale production testbed. On the single-rack testbed, we can conduct controlled experiments such as simulating AC failures to extensively evaluate our system. The production testbed allows us to evaluate our system under realistic, long-term computation workloads.

7.1. Single-Rack Testbed Experiments

Figure 2(a) shows the server groups and the temperature sensor locations on the rack of single-rack testbed. Five server groups, denoted Group 1 to Group 5, are controlled to run in either idle state (about 2% CPU utilization) or full utilization (about 90% CPU utilization). These settings are consistent with many data centers where servers running computational-intensive batch jobs tend to use all available CPUs [Moore et al. 2005]. We conduct various controlled experiments by adjusting servers' CPU utilization to simulate the normal running state of data centers, as well as turning off the cooling function of the AC to simulate a thermal emergency.

7.1.1. Prediction under Dynamic Workloads. The first experiment evaluates the performance of our system in response to CPU utilization changes. A total of 25 hours of data were collected during 6 days. Because the infrastructure ceiling vent is regularly shut down every night, we concatenate the data collected on different days when the ceiling

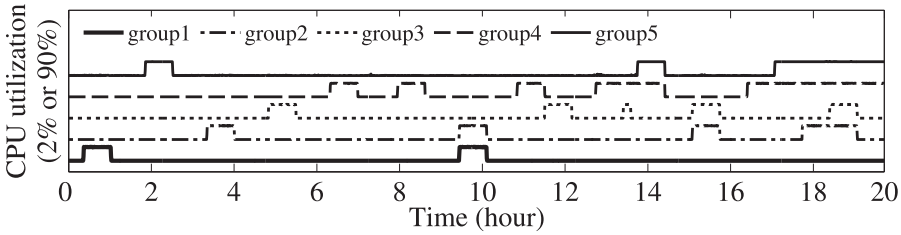

Fig. 6. CPU utilization of the training data.

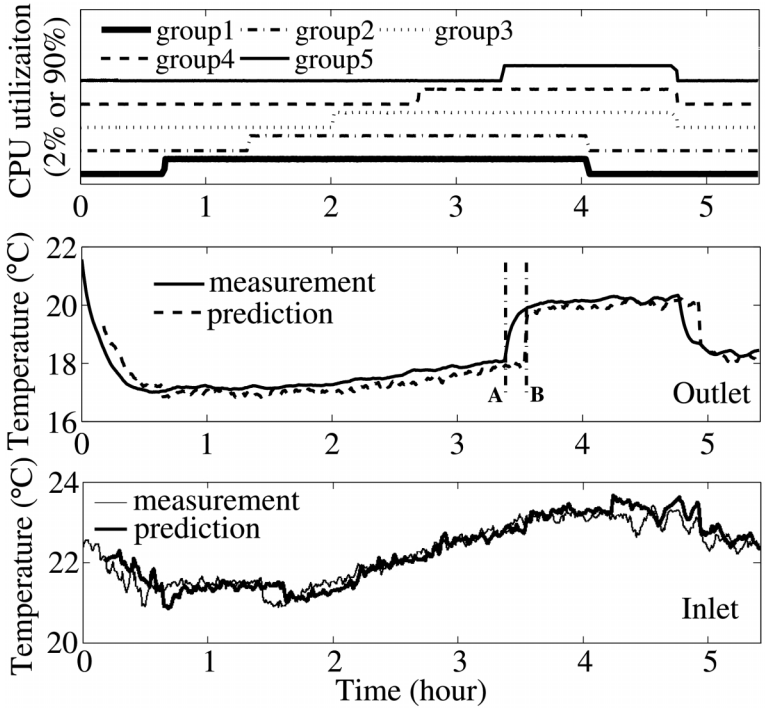

Fig. 7. Top: CPU utilization of test data. Middle: temperature measurements and predictions at an outlet of Group 5 with a 10-minute prediction horizon. Bottom: Temperature measurements and predictions at an inlet of Group 3 with a 10-minute prediction horizon.

vent is running. We use the first 20 hours of data as training data and the remaining 5 hours of data for testing. The settings of the prediction model include $R = 1$ and $k = 10$ min. Figure 6 shows CPU utilization of the training data, and Figure 7 shows the CPU utilization and temperature prediction at both inlets and outlets. We can see that our system can accurately predict the temperatures. From the middle graph of Figure 7, at about 30 minutes from the start, the temperature reached equilibrium as the start of our experiment. In the first 3 hours, because only Group 1 to Group 4 changed their running states, the measurements of Sensor 2 at an outlet of Group 5 did not change significantly. A small temperature rise during this period was caused by the complex airflow at the back of the rack. When the servers in Group 5 changed to full utilization during the 4th hour, a significant temperature rise is observed. With a 10-minute prediction horizon, each point on the dashed curve is calculated using measurements of all sensors 10 minutes earlier. We can see that the temperature at

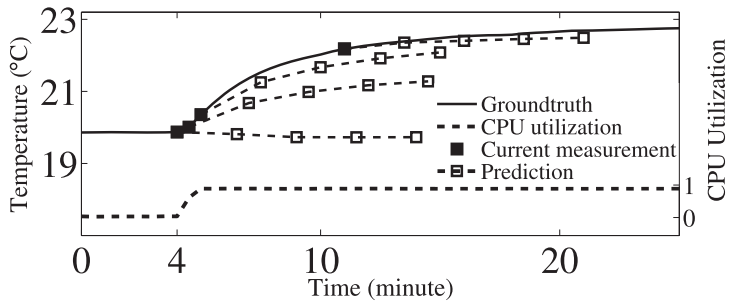

Fig. 8. Temperature evolution prediction. Each solid rectangle represents the temperature measurement at current time instance, and the white rectangles are the predicted temperatures at four different prediction horizons (0.5, 2.5, 5, and 7.5 minutes).

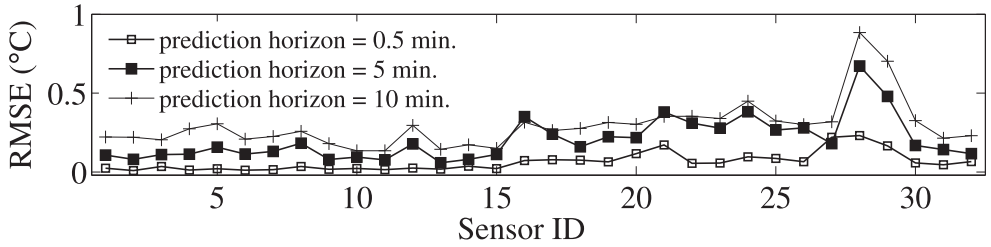

Fig. 9. Root-mean-square error (RMSE) of multihorizon temperature prediction.

the future time instant B is accurately predicted at the actual time instant A when the system observes the CPU utilization change. Although the prediction results well match the sensor measurements during the first 3 hours, we observe a considerable gap between the predicted temperatures and sensor measurements for a duration of 10 minutes (i.e., between A and B shown in the figure) after the CPU utilization change of Group 5. This is due to the fact that the system is not aware of the state change of Group 5 at time instance A. In this article, this type of error is referred to as a *horizon-induced* prediction error. According to the multihorizon prediction scheme discussed in Section 7.1.2, the duration that suffers horizon-induced prediction error can be shortened by setting a smaller prediction horizon. This hypothesis is verified in Section 7.1.6. Different from the temperature at the outlets, the temperature at the inlets is mainly affected by the complex heat recirculation. The bottom graph shows that our system can also accurately predict the temperature at the inlet. During the 5-hour testing period, the average absolute prediction error over all sensors is only 0.3°C.

7.1.2. Multihorizon Prediction. In our prediction system, by training models with different k in Equation (1), we can build multiple models to predict the evolution of temperature in the future. Figure 8 shows the results of different prediction horizons of 0.5, 2.5, 5, and 7.5 minutes. At about the 4th minute, when the CPU utilization just begins to increase, the predicted temperatures at different prediction horizons are similar to the current measurement. After the system evolved to the second solid rectangle, where the CPU utilization had increased significantly from 2% to 90%, the system predicts an increasing trend of temperature evolution for the following four horizons. From the time instance of the 3rd solid rectangle, the predicted temperature evolution starts to match the groundtruth. Figure 9 shows the RMSE of multihorizon predictions for each sensor. We can see that the RMSE generally increases with the prediction horizon.

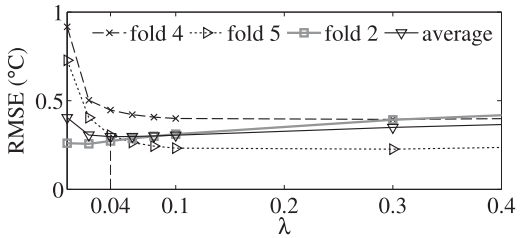

Fig. 10. RMSE of prediction versus λ in cross-validation experiments.

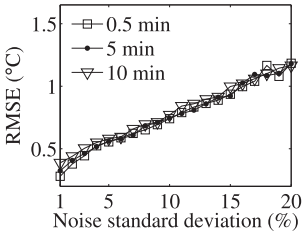

Fig. 11. RMSE under noisy data.

This conforms to the intuition that the temperature at a further time instance in the future is less correlated with historical measurements in a dynamic environment. The RMSEs are less than 0.5°C for most sensor locations. Slightly larger RMSEs are observed at sensor 28 and 29. We found that this is caused by the slight displacement of the two sensors during the experiment. Nevertheless, the RMSEs are still less than 1°C.

7.1.3. Effectiveness of Regularized Regression. In this experiment, we conduct cross-validation based on the 14-hour dataset used in Section 7.1.2 to evaluate the effectiveness of the regularized regression discussed in Section 5.2.1. Performance evaluation of model training depends on how the dataset is partitioned into training and test data. Ten-fold cross-validation [Ye 2003, p. 435] is commonly used to mitigate the impact of dataset partitioning in evaluating model training performance. Specifically, in each *fold*, we choose 1.4 hours of continuous data for testing and the remaining data for training. For each fold, given a regulation factor λ, we conduct model trainings and predictions with five different horizons from 0.5 minutes to 10 minutes. The average RMSE of prediction over all horizons is used to represent the average prediction error given λ. This process is repeated 10 times (i.e., 10 folds) with different partitions of training and test datasets. Figure 10 shows the average RMSE for three individual folds versus λ. In the experiment of fold 4, the average RMSE of prediction is as high as 1°C when the training is not regularized (i.e., $\lambda = 0$). The average RMSE decreases with λ when $\lambda < 0.3$ and exhibits a slight increase after that. The experiment of fold 5 shows a similar trend, with much lower RMSE. In the experiment of fold 2, the average RMSE always increases with λ. Figure 10 also shows the average RMSE over all folds, which characterizes the expected performance given any training/test data partition. From the average RMSE over all folds shown in Figure 10, we can see that in a large range of λ (i.e., from 0 to 0.4), the regularized regression outperforms the unregularized version (i.e., $\lambda = 0$). Note that the typical setting of λ is no greater than 1. Moreover, when $\lambda = 0.04$, the average RMSE is minimized and reduced by 25% with respect to the unregularized result. Under this setting, the trained models yield good prediction accuracy across all folds. Therefore, 0.04 is a desirable setting for λ for our single-rack testbed.

7.1.4. Performance under Noisy Sensor Measurements. In this section, we evaluate the prediction performance under noisy sensor measurements with dynamic CPU utilizations. In particular, we manually add Gaussian white noise to the sensor measurements. The standard deviation of noise is proportional to the range of its readings in the data traces. Figure 11 shows the RMSE of predictions versus the increasing noise standard deviation for each thermal variable under different prediction horizons. Consistent with intuition, the RMSE increases with the noise level. However, even with noise standard deviation of up to 15% of the sensor reading range, the RMSEs are still within 1°C. In

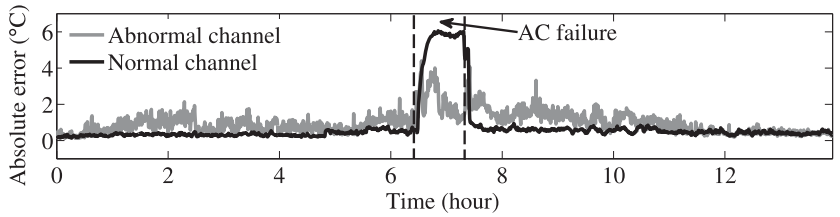

Fig. 12. Average absolute temperature prediction error (prediction horizon = 5 minutes).

practice, various noise suppression techniques (e.g., moving window average) can be employed to mitigate sensor measurement errors and improve prediction accuracy.

7.1.5. Multichannel Prediction. In this experiment, we evaluate the accuracy of prediction in multiple thermal conditions (i.e., channels). Because AC malfunction is a major cause of server overheating in data centers, we conducted a controlled experiment to simulate the AC failure on our single-rack testbed. We construct two channels corresponding to the normal running state and AC failure, respectively. A total of 10 hours of data were collected while the servers ran in a normal state with different CPU utilization combinations. These data are used to train the normal channel of the prediction system. The prediction horizon is set to 5 minutes. Then, another 14 hours of data, which contain both normal running state and AC failure, were collected. A transient CFD simulation is conducted using the sensor data (after excluding the temperature measurements at server inlets/outlets) collected during this 14-hour experiment. The CFD-simulated training data, together with the 10 hours of real measurements in normal running state, are then used to train the channel of AC failure. In real data centers, it is often infeasible to collect training data for the scenario of AC failure. Therefore, to ensure the realism of our experiments, we did not use the sensor measurements during AC failure to calibrate the CFD.

Figure 12 shows the absolute prediction errors of the two channels with respect to the groundtruth sensor measurements. The system exhibits a very small absolute error in the normal state, whereas it suffers from up to 6°C absolute error during AC failure. This is because the training data for the normal channel do not capture this abnormal situation. On the contrary, the AC failure channel exhibits a slightly higher absolute error than the normal channel during the normal state, whereas it has a significantly lower absolute error during the AC failure. From this result, we can see that the simulated training data generated by CFD can help the real-time prediction model capture various thermal emergencies. In practice, several different abnormal channels can be constructed with CFD according to the possible cooling system failure situations. The detection results from different channels can further be fused using existing data fusion techniques [Varshney 1996].

7.1.6. Sufficiency of Traning Data from CFD. In this set of experiments, we evaluate the training data generation approach described in Section 5.2.2. These experiments focus on the thermal emergency of AC failure. We first explain how we generate the training data traces. Figure 13 shows the temperature trace of a server inlet in one of the transient simulations. The simulation starts with all servers in idle state with 0% CPU utilization, followed by a change of CPU utilization at about the 4th minute. Following the random approach described in Section 5.2.2, the new CPU utilization of a server group is randomly drawn from a uniform distribution over [0%, 100%]. The inlet temperature shown in Figure 13 starts to rise because of air recirculation. The simulated AC failure occurs at about the 14th minute, and the AC recovers from failure at about the 24th minute. We conduct 50 transient simulations to generate

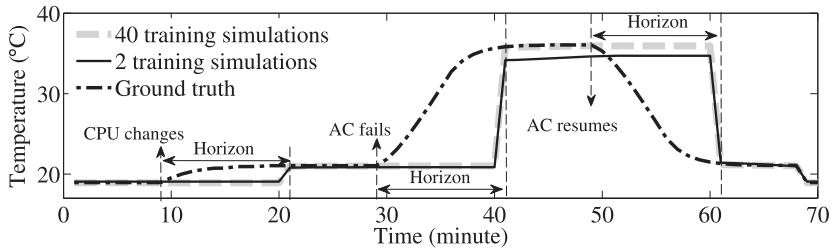

Fig. 13. Example of training data generated from CFD for an AC failure emergency. The simulation starts with all the servers in idle status, followed by a uniform random change on CPU utilization at about the 4th minute. Then, the AC fails at about the 14th minute and resumes at about the 24th minute.

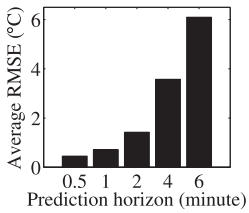

Fig. 14. Horizon-induced prediction error.

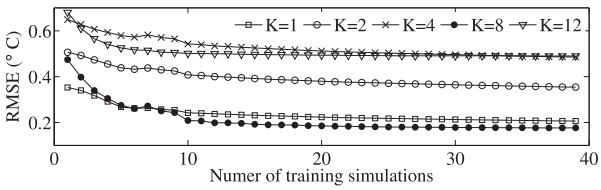

Fig. 15. Prediction errors with incremental training samples.

the training data, in which the CPU utilization of each server group is drawn from the uniform distribution. Moreover, we conduct two transient simulations with extreme conditions; that is, the CPU utilization of all server groups is either 100% or 0%.

As discussed in Section 5.2.2, the number of generated data traces should be large enough to prevent overfitting. We design an experiment to evaluate the impact of the amount of training data on the performance of model training. We choose 10 out of 50 transient simulations as the test data. Initially, only the two transient simulations with extreme conditions are included in the training dataset. We then incrementally add a simulation that is chosen from the unused simulations to the training dataset. For each training dataset, we evaluate the prediction performance of the trained model using the test data of 10 transient simulations. When we compute the RMSE to characterize the prediction error, we carefully choose the testing results to exclude the horizon-induced prediction error, which is explained in Section 7.1.1. For instance, we exclude the durations labeled by "Horizon" in Figure 13, which suffer from horizon-induced errors. The root cause of this type of errors (as shown in Figure 14) is the horizon, rather than the insufficiency of training data.

Figure 13 shows two traces of prediction when 2 and 40 transient simulations are used as training data, respectively. Figure 15 shows the prediction error versus the size of the training dataset (i.e., the number of transient simulations) under various settings of prediction horizon. We can see that the prediction error generally decreases with the size of the training dataset. In particular, when more than 10 transient simulations are used to train the model, the prediction error becomes flat. This result shows that, by our training data generation approach, a small number of transient simulations can be sufficient to train the model. Moreover, from Figure 15, we do not see strong correlation between the prediction error and horizon. This is because, after excluding the durations suffering from horizon-induced prediction error, the remaining durations are mostly steady states in which horizon is not a major factor.

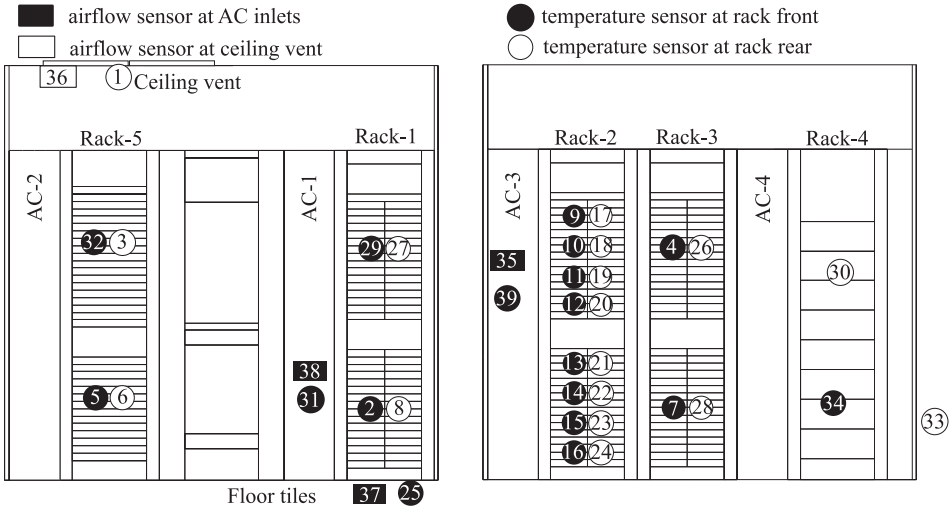

Fig. 16. Front view of the two rows of racks, which face each other in the server room.

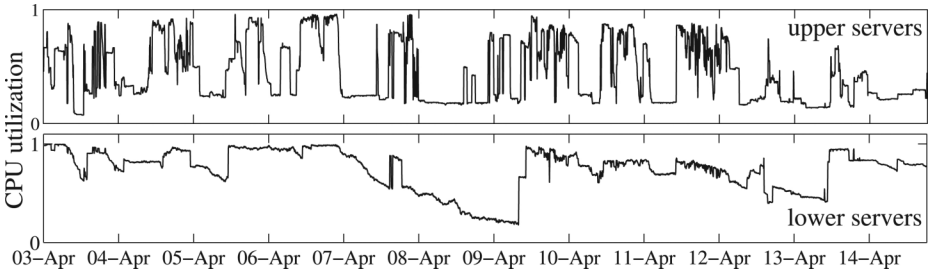

Fig. 17. CPU utilization of servers on upper and lower levels of Rack-2.

7.2. Production Testbed Experiments

We also deployed and evaluated our system on a small-scale production testbed in a server room of the HPCC at Michigan State University. In this testbed, we deployed 35 temperature sensors and four airflow velocity sensors. Figure 16 shows the sensor deployment from the front view of the two rack rows. To evaluate the impact of sensor density, we deploy 16 sensors on one rack (Rack-2) while other racks are instrumented with two or four sensors. Differing from the single-rack testbed whose CPU utilization is controlled, the CPU utilization in the HPCC testbed is subject to real use and is thus dynamic. Therefore, accurate temperature prediction is more challenging. Figure 17 shows the average CPU utilization of the upper and lower section of the servers on Rack-2 in a 12-day period. We can observe that the lower section of servers usually has high CPU utilization, except on April 8, which is a Sunday. On the contrary, the upper section of servers has more variable CPU utilization. In this section, we evaluate our prediction approach in the HPCC testbed using data collected over the course of 15 continuous days. The data from the first three days (March 31–April 2, 2012) are used as training data, whereas data from the following 12 days are used for prediction evaluation.

7.2.1. Dimension Reduction. On our single-rack testbed described in Section 7.1, the dimension of the state $\mathbf{p}$ is 64. However, on the production data center testbed, the

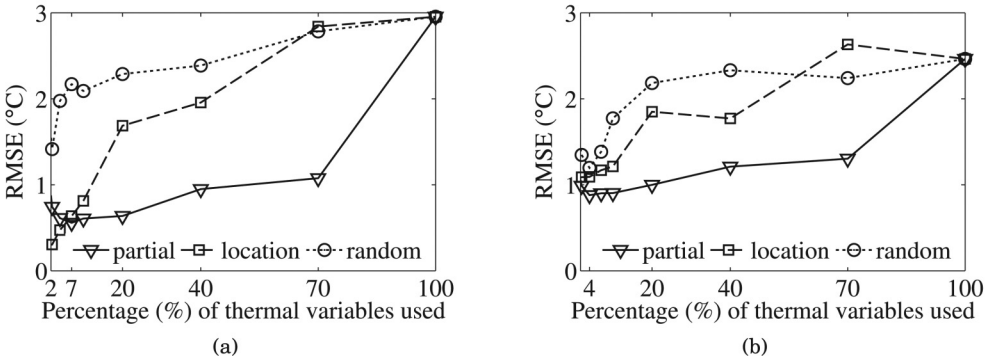

Fig. 18. Prediction error versus the percentage of selected variables in dimension reduction. (a) All data are used for testing; (b) only transient data are used for testing.

dimension of the state $\mathbf{p}$ grows to 229. As discussed in Section 5.3, it is desirable to perform dimension reduction to avoid overfitting. In addition to the *Partial Correlation* described in Section 5.3, we employ two baseline approaches. The first baseline approach, referred to as *Random*, randomly selects variables. The second baseline approach, referred to as *Location*, selects the variables that are geographically closest to the monitored location. In the following experiments, we vary the percentage of selected variables over all available variables.

Figure 18(a) shows the average RMSE of the prediction results with prediction horizons of 5 and 10 minutes. For the *Partial Correlation*, RMSE drops from 3°C to about 0.6°C when the percentage of selected variables reduces from 100% to 7%. This result conforms to our motivation for dimension reduction discussed in Section 5.3. However, when the percentage continues to decrease, RMSE starts to increase since the number of variables is too small to contain enough information for a good prediction. When the percentage reduces from 100% to 7%, the RMSE of *Location* is larger than that of the *Partial Correlation*. However, the *Location* outperforms the *Partial Correlation* when the percentage is lower than 7%. Under the *Location* approach, the monitored temperature itself is assigned with a very high weight in $\mathbf{a}_j$ given by Equation (2) when a small percentage of variables are chosen. Therefore, the resulted prediction models tend to follow an autoregression model. As a result, when the temperatures are in steady state, the autoregression prediction is highly accurate. Because the test data used for Figure 18(a) includes many steady states, the *Location* yields good performance when the percentage is low. However, such an autoregression prediction model is not helpful for predicting overheating in emergencies, such as AC failures, because the variables related to AC may not be chosen by the *Location*. However, it is unlikely to create these thermal emergencies in the production data center for system training. As a compromise, we manually select transient states, which are mainly caused by the changes of CPU utilization, as the test data. With these transient test data, the *Partial Correlation* consistently outperforms the *Location*, as shown in Figure 18(b). The *Partial Correlation* achieves minimal RMSE when 4% of variables are selected. Moreover, from both Figure 18(a) and Figure 18(b), the *Partial Correlation* achieves low RMSE in a wide range of settings (2–70%), which allows flexible setting without sacrificing the prediction performance substantially. For the experiments conducted in the rest of this article, we perform dimension reduction using *Partial Correlation* with the setting of 4%. From Figure 18(a) and Figure 18(b), *Random* consistently yields the worst performance.

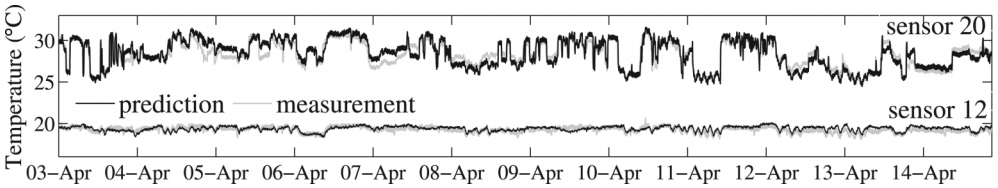

Fig. 19. Long-term monitoring with a 10-minute prediction horizon. Sensor 20 and sensor 12 are located at server outlet and inlet, respectively.

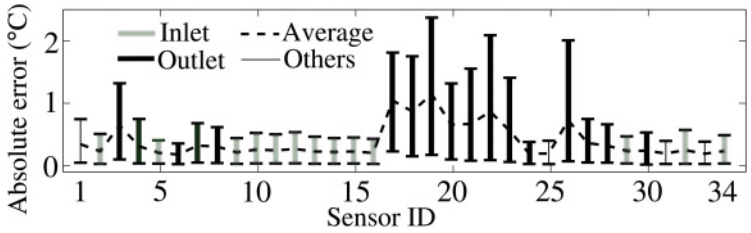

Fig. 20. Absolute errors with the 90% error bound for each sensor with 10-minute prediction horizons.

7.2.2. Long-term Monitoring. Figure 19 shows the prediction results at two locations during 12 days. The prediction horizon is set to 10 minutes. We can observe that our prediction results well match the groundtruth measurements of both server inlet and outlet sensors. Sensor 20, located at a server outlet, exhibits slightly larger prediction errors. This is because the server outlets suffer more influence from system workloads and hence have more dynamic thermal profiles. Figure 20 shows the average absolute prediction error and the 90% error bound for all sensor locations. We observe that the prediction errors on outlet sensors are slightly higher than on inlet sensors. Nevertheless, the average absolute error of outlet predictions is only around 1°C, and 90% of predictions have errors lower than 2°C. We also evaluate the prediction errors under different prediction horizon settings. Figure 21 shows the empirical Cumulative Distribution Function (CDF) of prediction error over all sensor locations. Similar to the results in Figure 10, the prediction error increases with the prediction horizon.

7.2.3. CFD-Assisted Prediction. In this section, we evaluate the performance of using CFD to assist temperature predictions. We focus on evaluating how effectively CFD modeling reduces the number of required sensors under the normal running state because no thermal emergencies were observed on the production testbed during the 15-day experimental period. Specifically, during the model training, we remove the measurements of some sensors and replace them with CFD transient simulation results. The removed temperature sensors will not be selected as thermal variables. However, with their CFD replacements, they can be used as output to train the temperature predictions at those locations. We use the first 3 days of boundary condition data (e.g., CPU utilization) to drive the CFD transient simulation. Then, the sensor data of the first day are used to construct the calibration functions discussed in Section 4.3. After that, all the 3-day simulated training data are calibrated using the calibration functions. Figure 23 shows a significant accuracy improvement after CFD calibration.

We evaluate the performance of the CFD-assisted prediction by gradually removing sensors in the model training. As shown in Table I, we gradually replace measurements of some sensors with CFD simulation results. In Case 1, all 39 sensors are used for training. Then, we replace sensor 10, 12, . . . , 24 from Case 1 to generate Case 2, which uses 31 sensors in total for training. In Case 5, 26 sensors (i.e., 67% of all sensors) are replaced with data generated from CFD. The empirical CDF of absolute errors

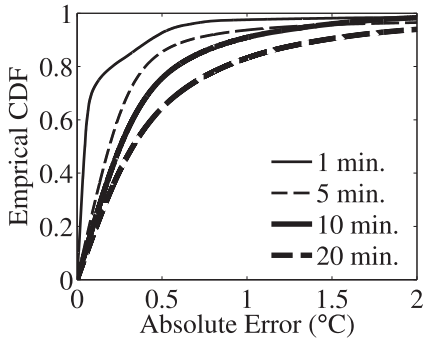

Fig. 21. Empirical CDF of absolute error for all sensors with different prediction horizons.

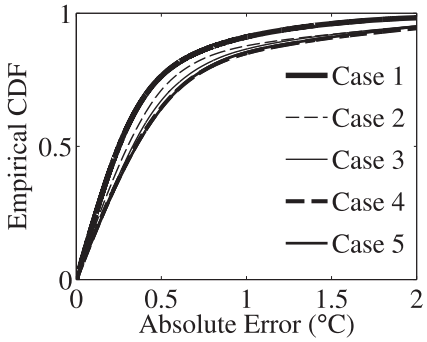

Fig. 22. Empirical CDF of absolute error when different subsets of sensors are used in model training.

Table I. Evaluation Scheme of Replacing Sensors with CFD

Case	Sensors Removed	Total
1	None	39
2	10, 12, 14, 16, 18, 20, 22, 24	31
3	32, 3, 29, 27, 4, 26	25
4	9, 17, 13, 21, 36, 1	19
5	5, 6, 25, 37, 7, 28	13

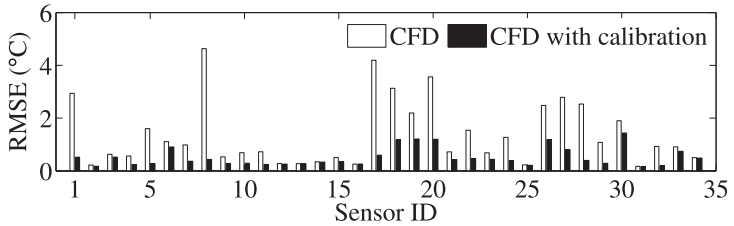

Fig. 23. RMSE of CFD calibration in production testbed.

of different cases is plotted in Figure 22. The prediction horizon is 10 minutes. We see that prediction accuracy increases with the number of sensors. This is consistent with the intuition that the CFD data is not as accurate as actual sensor measurements. However, it can be seen that, even when fewer than 40% of sensors are deployed in Case 5, 85% of predictions have absolute errors lower than 1°C. Overall, our approach can remove 67% of sensors for monitoring the inlets and outlets of all servers while only increasing the average prediction error by 0.2°C. For all cases, the average RMSEs are less than 1°C, whereas the maximum RMSE is within 2°C. This result clearly demonstrates the advantage of integrating calibrated transient CFD modeling with real sensor measurements. Currently, the sensor reduction is performed empirically. The number of required sensors to achieve a certain prediction accuracy is highly affected by physical properties in the data center. This is still an open issue and left for our future work.

8. CONCLUSION AND FUTURE WORK

In this article, we describe the design and implementation of a novel cyber-physical system for predicting the temperature distribution of data centers. Our approach integrates CFD modeling and real-time data-driven prediction to achieve high-fidelity

temperature forecasting in various thermal conditions found in data centers, including rare but critical thermal emergency situations like AC failures. We implemented the system on a single-rack testbed and a testbed of five racks and 229 servers in a production high-performance computing center. Extensive experimental results show that our approach can accurately predict temperatures up to 10 minutes into the future, even in the presence of highly dynamic server workloads.

A key advantage of our approach is to leverage those CFD simulation models that are already available for many production data centers. However, the CFD models created for large-scale data centers typically have a coarse granularity and considerable errors. In the future work, we will evaluate the impact of CFD accuracy on temperature forecasting fidelity in large-scale data centers. In addition, we will study thermal actuation mechanisms that can control server workloads and cooling systems based on the predicted temperature evolution.

REFERENCES

Active Power, Inc. 2007. Data center thermal runaway: A review of cooling challenges in high density mission critical environments. http://new.activepower.com/documents/white_papers/.

Aperture Research Institute. 2007. Data center professionals turn to high-density computing as major boom continues. http://www.emersonnetworkpower.com/en-EMEA/Brands/Aperture/ApertureResearch Institute/Research/Documents/ari_high_density_4_01_07.pdf.

ASHRAE Technical Committee 9.9. 2011. 2011 thermal guidelines for data processing environments—Expanded data center classes and usage guidance. http://ecoinfo.cnrs.fr/IMG/pdf/ashrae_2011_thermal_guidelines_data_center.pdf.

Cullen Bash and George Forman. 2007. Cool job alloction: Measuring the power savings of placing jobs at cooling-efficient locations in the data center. In *Proceedings of USENIX Annual Technical Conference (USENIX)*. 1–6.

Cullen E. Bash, Chandrakant D. Patel, and Ratnesh K. Sharma. 2006. Dynamic thermal management of air cooled data centers. In *Proceedings of the 10th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronics Systems (ITherm)*.

Geoffrey C. Bell. 2012. Improving data center efficiency with rack or row cooling devices: Results of "Chill-Off 2" comparative testing. *Federal Energy Management Program.* http://datacenters.lbl.gov/sites/all/files/dc_chilloff2.pdf.

Susmit Biswas, Mohit Tiwari, Timothy Sherwood, Luke Theogarajan, and Frederic T. Chong. 2011. Fighting fire with fire: Modeling the datacenter-scale effects of targeted superlattice thermal management. In *Proceedings of the 38th International Symposium on Computer Architecture*. 331–340.

Jinzhu Chen, Rui Tan, Guoliang Xing, and Xiaorui Wang. 2014. PTEC: A system for predictive thermal and energy control in data centers. In *Proceedings of the 35th IEEE Real-Time Systems Symposium (RTSS)*. 218–227.

Jeonghwan Choi, Youngjae Kim An, Jelena Srebric, Qian Wang, and Joonwon Lee. 2007. Modeling and managing thermal profiles of rack-mounted servers with thermostat. In *Proceedings of the 13th International Symposium on High-Performance Computer Architecture*. 205–215.

Supasate Choochaisri, Vit Niennattrakul, Saran Jenjaturong, Chalermek Intanagonwiwat, and Chotirat Ann Ratanamahatana. 2010. SENVM: Server environment monitoring and controlling system for small data center using wireless sensor network. http://arxiv.org/pdf/1105.6160.pdf.

Degree Controls, Inc. 2011. *F333 Airflow Sensor User Guide*.

Nosayba El-Sayed, Ioan Stefanovici, George Amvrosiadis, Andy A. Hwang, and Bianca Schroeder. 2012. Temperature management in data centers: Why some (might) like it hot. In *Proceedings of the 12th ACM Sigmetrics/Performace Joint International Conference on Measurement and Modeling of Computer Systems.* 163–174.

Emerson Network Power. 2011. State of the Data Center. Retrieved from http://www.emersonnetworkpower.com/.

Taliver Heath, Ana Paula Centeno, Pradeep George, Luiz Ramos, Yogesh Jaluria, and Ricardo Bianchini. 2006. Mercury and freon: Temperature emulation and management for server systems. In *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. 106–116.

Arther E. Hoerl and Robert W. Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 1 (1970), 55–67.

Michael Jonas, Rose Robin Gilbert, Joshua Ferguson, Georgios Varsamopoulos, and Sandeep Gupta. 2012. A transient model for data center thermal prediction. In *Proceedings of the 3rd International Green Computing Conference (IGCC)*. 1–10.

Jon Lenchner, Canturk Isci, Jeffrey Kephart, Christopher Mansley, Jonathan Connell, and Suzanne McIntosh. 2011. Toward data center self-diagnosis using a mobile robot. In *Proceedings of the 8th IEEE International Conference on Autonomic Computing (ICAC)*. 81–90.

Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, and David Culler. 2005. TinyOS: An operating system for sensor networks. *Ambient Intelligence* (2005), 115–148.

Lei Li, Chieh-Jan Mike Liang, Jie Liu, Suman Nath, Andreas Terzis, and Christos Faloutsos. 2011. ThermoCast: A cyber-physical forecasting model for data centers. In *Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 1370–1378.

Chieh Jan Mike Liang, Jie Liu, Liqian Luo, Andreas Terzis, and Feng Zhao. 2009. RACNet: A high-fidelity data center sensing network. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 15–28.

Chris Mansley, Jonathan Connell, Canturk Isci, Jonathan Lenchner, Jeffrey O. Kephart, Suzanne McIntosh, and Michael Schappert. 2011. Robotic mapping and monitoring of data centers. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 5905–5910.

Memsic Corp. 2012. TelosB, Iris datasheets. http://www.memsic.com/wireless-sensor-networks/.

Justin Moore, Jeff Chasey, and Parthasarathy Ranganathanz. 2006. Weatherman: Automated, online, and predictive thermal mapping and management. In *Proceedings of the 3rd IEEE International Conference on Autonomic Computing (ICAC)*. 155–164.

Justin Moore, Jeff Chasey, Parthasarathy Ranganathanz, and Ratnesh Sharmaz. 2005. Making scheduling "Cool": Temperature-aware workload placement in data centers. In *Proceedings of the USENIX Annual Technical Conference (USENIX)*. 5.

John Niemann. 2006. Best practices for designing data centers with the InfraStruXure InRow RC. *Application note of American Power Conversion*. http://www.apcmedia.com/salestools/JNIN-6N7SRZ/JNIN-6N7SRZ_R0_EN.pdf?sdirect=true.

Luiz Ramos and Ricardo Bianchini. 2008. C-Oracle: Predictive thermal management for data centers. In *IEEE 17th International Symposium on High Performance Computer Architecture (HPCA)*. 111–122.

Neil Rasmussen. 2011. Cooling options for rack equipment with side-to-side airflow. http://www.apcmedia.com/salestools/SADE-5TNRKJ/SADE-5TNRKJ_R1_EN.pdf?sdirect=true.

Umesh Singh, Amarendra K. Singh, S. Parvez, and Anand Sivasubramaniam. 2010. CFD-based operational thermal efficiency improvement of a production data center. In *Proceedings of the 1st USENIX Workshop on Sustainable Information Technology (SustainIT)*. 6.

Alan Stuart, Keith Ord, and Steven Arnold. 2009. *Kendall's Advanced Theory of Statistics: Classical Inference and the Linear Model* (6 ed.). John Wiley & Sons.

Qinghui Tang, Sandeep K. S. Gupta, and Georgios Varsamopoulos. 2008. Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach. In *IEEE Transactions on Parallel and Distributed Systems*, 19: 1458–1472.

Qinghui Tang, Tridib Mukherjee, Sandeep K. S. Gupta, and Phil Cayton. 2006. Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters. In *Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing (ICISIP)*. 203–208.

U.S. Environmental Protection Agency. 2007. *Report to Congress on Server and Data Center Energy Efficiency*. http://hightech.lbl.gov/documents/data_centers/epa-datacenters.pdf.

L. J. P. Van der Maaten, E. O. Postma, and H. J. Van Den Herik. 2009. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research* 10 (2009), 1–41.

P. K. Varshney. 1996. *Distributed Detection and Data Fusion*. Springer.

Xiaodong Wang, Xiaorui Wang, Guoliang Xing, Jinzhu Chen, Cheng-Xian Lin, and Yixin Chen. 2011. Towards optimal sensor placement for hot server detection in data centers. In *Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS)*. 899–908.

John F. Wendt (Ed.). 1995. *Computational Fluid Dynamics—An Introduction* (3rd ed.). Springer.

WikiMedia Foundation. 2010. Global outage (cooling failure and DNS). http://blog.wikimedia.org/2010/03/24/global-outage-cooling-failure-and-dns/.

Nong Ye (Ed.). 2003. *The Handbook of Data Mining*. Lawrence Erlbaum Associates.

Received February 2013; revised December 2013; accepted June 2014