2012 IEEE 33rd Real-Time Systems Symposium

A High-Fidelity Temperature Distribution Forecasting System for Data Centers

Jinzhu Chen[1]; Rui Tan[1]; Yu Wang[1]; Guoliang Xing[1]; Xiaorui Wang[2]; Xiaodong Wang[2]; Bill Punch[1]; Dirk Colbry[1]
[1]Michigan State University, USA; [2]Ohio State University, USA

Abstract—**Data centers have become a critical computing infrastructure in the era of cloud computing. Temperature monitoring and forecasting are essential for preventing overheating-induced server shutdowns and improving a data center's energy efficiency. This paper presents a novel cyber-physical approach for temperature forecasting in data centers, which integrates Computational Fluid Dynamics (CFD) modeling, *in situ* wireless sensing, and real-time data-driven prediction. To ensure the 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 significantly reduces the computational complexity of online temperature modeling and prediction, which enables a portable, noninvasive thermal monitoring solution that does not rely on the infrastructure of monitored data center. We extensively evaluated our system on a rack of 15 servers and a testbed of five racks and 229 servers in a 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.**

I. 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 [1]. For instance, Wikipedia, a popular online encyclopedia, went down on March 24th, 2010 because of server overheating [2]. Currently, the common practice to prevent overheating is to overcool the server rooms. Due to such a conservative strategy, the cooling systems consume excessive power, which takes up to 50% of the total energy consumption in many data centers [3].

Various thermal management schemes for improving the energy efficiency of data centers rely on real–time and high–fidelity temperature monitoring [4] [5] [6] [7]. Recently, Wireless Sensor Networks (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, the 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.

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 designed based on simplified physical and cyber models would not yield satisfactory performance. Second, the number of locations where the 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 of such locations. Third, it is desirable to decouple the prediction system and the computing resources of the monitored data center. This design not only avoids the potential interruptions to the prediction system due to unexpected server shutdowns, but also improves the system portability. To this end, the prediction system must operate on limited computing resources while maintaining high prediction fidelity.

To address the above challenges, we propose a novel cyber–physical approach that integrates *in situ* wireless sensors, transient Computational Fluid Dynamics (CFD) modeling [8] and real–time data–driven prediction algorithms. CFD is a widely adopted numerical tool that can simulate the future evolution of temperature distribution of data centers. However, without accounting for runtime behaviors of the data center, CFD often yeilds 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. Sec–

1052-8725 2012
U.S. Government Work Not Protected by U.S. Copyright
DOI 10.1109/RTSS.2012.73

IEEE computer society

ond, 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, as CFD can simulate the temperature at the uninstrumented locations, our approach can potentially reduce the number of sensors required.

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 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.

II. Related Work

Existing approaches for data center thermal management can be broadly divided into two groups. The first group of approaches focuses on the assignment of server workload based on physical thermodynamic models to improve the energy efficiency of data centers. In [5], cooling-efficient servers are identified and assigned with more workload. In [6], heat recirculation is minimized by distributing computing power to the servers with less heat recirculation at their inlets. Tang *et al.* propose abstract models to distribute computing power in data center by minimizing peak inlet temperatures [7]. The second group of approaches focuses on the prediction of temperature distribution to prevent thermal emergencies. In [9] [10], the temperature distribution of a single server is emulated based on simplified thermodynamic laws, CPU temperature/utilization, and airflow velocity. In [11], a heat flow model is proposed to characterize the heat recirculation and predict the temperature distribution. In [12], artificial neural network is employed to learn and predict the steady-state temperature distribution under static workload assignment. However, these approaches rely on steady-state thermal models, which cannot well model the temperature evolution when the heat dissipation from servers is dynamic. They also require a controlled training procedure which is usually intrusive or even infeasible to 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 [13], 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 the airflow dynamics, which 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 row of the racks [14] [15] or near the racks [16], which generates significant side-to-side airflow.

Several sensor systems have been developed for temperature monitoring in data centers [17] [18]. RACNet [17] is designed for reliable data collection in large-scale data centers, where each node is connected with multiple daisy-chained temperature sensors. In [19], 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 map [20] and energy management [21]. However, these studies are not concerned with the real-time temperature prediction.

III. Problem Statement and Approach Overview

A. 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 is required to be within a small range (e.g., 15°C to 27°C [22]). The outlet temperature characterizes the amount of heat that needs to be removed by the 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 amount of heat over time. The heat is dissipated by the extremely complex airflows, which 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 1°C error bound. This requirement ensures that the predicted temperature will not trigger excessive false alarms of overheating or miss real overheating events. Moreover, as shown in [6], an 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 in a considerably long time duration (e.g., 10 minutes), referred to as *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 [23] in the 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 the data center administrators to take necessary intervention. 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 the 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 requirement. The overhead of the prediction system should be affordable to low-end servers, desktop computers or even embedded computing devices, such that the system can be easily deployed and operated in a noninvasive fashion, without relying on the infrastructure of the monitored data center.

Computational Fluid Dynamics (CFD) [8] is a widely used tool to guide the 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 [19] [24]. This often leads to highly conservative temperature setpoints, resulting in excessively high power consumption of 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.

B. 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 the modeling fidelity, multiple models are used to capture the normal and various abnormal working states such as the failure of different AC units. A challenge of such training-based approach is to collect sufficient data sets that cover various thermal conditions, especially for the 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 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, significantly reducing the efforts of CFD model tuning. Another advantage of our approach 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 the production data centers.

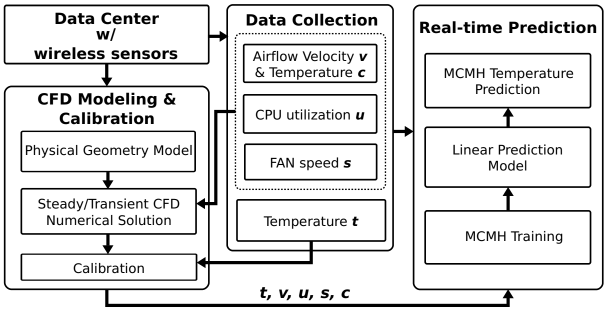
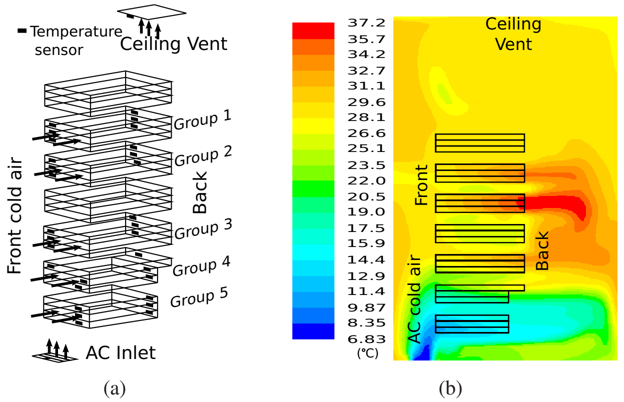


Figure 1. Prediction system architecture.

Fig. 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 wireless sensor network to collect the measurements of temperatures and airflow velocities. The historical measurements are used to calibrate the CFD modeling and train the prediction models, while the runtime sensor measurements are fed to the real-time prediction component to predict temperatures. (2) ***CFD modeling and calibration***. In addition to the sensor measurements, a key feature of our system is to leverage the transient CFD simulation to compute the 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 the training data to the real-time prediction component. (3) ***Real-time multi-channel and multi-horizon temperature prediction***. The real-time prediction component constructs time series prediction models with training data from both historical measurements and CFD simulations,



(a) (b)

Figure 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.

and outputs the runtime temperature predictions. Although complex non-linear models may achieve good prediction accuracy, they often have high complexity. Our solution uses multiple simple linear models to approximate the complex non-linear thermodynamic laws. For each different major thermal condition (hereafter referred to as *channel*), such as the failure of different AC units, multiple prediction models with different prediction horizons are constructed. Different prediction horizons of our system give administrators more flexibility to implement thermal actuators, e.g., taking appropriate measures in an incremental fashion.

IV. CFD Modeling and Calibration

In this section, we first briefly introduce the Computational Fluid Dynamics (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.

A. Background on CFD

CFD is a widely adopted numerical tool to simulate the future temperature evolution. It iteratively solves a system of fluid and heat transfer equations in the form of non-linear partial differential equations under the constraints of mass, momentum and energy conservation. The non-linear nature of these equations and the complex boundary conditions in data center environment (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 where all cells meet the convergence requirements, giving the steady-state temperature distribution. For the transient simulation, the model is also discretized into small time steps in 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) such that CFD can simulate any normal or abnormal thermal situations. Therefore, the accuracy of the transient CFD modeling is particularly important for achieving high prediction fidelity.

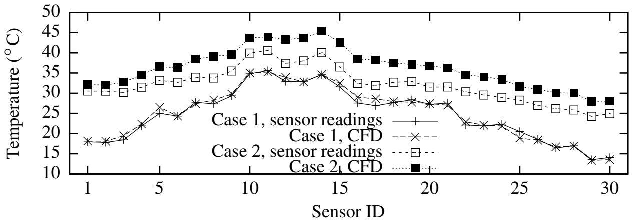


Figure 3. Real sensor readings and CFD prediction. Case 1: servers in Group 1 and Group 2 run with full utilization; Case 2: AC failure.

B. A Case Study

We now present a case study of using CFD to model a testbed of rack servers, which helps us understand the performance limitations of CFD. Fig. 2(a) shows the physical geometry of rack server testbed. A total of 15 servers on the rack are grouped into 5 server groups. The detailed settings of the server rack can be found in Section VI. 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 around the rack. Node 1 to 15 are installed at the outlets of the servers and Node 16 to 30 are installed at the inlets of the servers.

Fig. 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. Fig. 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 Fig.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 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-

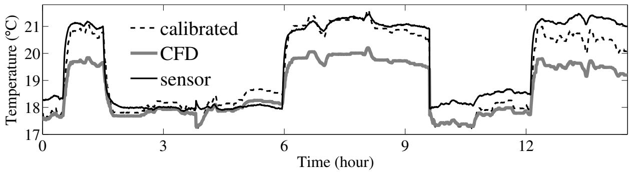


Figure 4. Transient temperatures at the outlet of the lowest server.

state temperature distribution for the case of AC failure (referred to as Case 2). Fig. 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 transient simulation, which is critical for the performance of real–time prediciton. Fig. 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 temperature at this sensor location. It can be clearly seen that CFD result contains significant biases with respect to the real sensor readings. The major reason of those errors is that CFD does not exactly model the true data center environment and all the 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 IV–C 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.

C. CFD Calibration

The results from Section IV–B show that the CFD simulation exhibits considerable errors, particularly in transient–state simulations. To address this limitation, we propose to calibrate 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 $a_{i,k}$'s are the coefficients to be learned from training data. The rationale for choosing this function is that the CFD temperature error is highly related to historical temperature. By providing real sensor data as y_i, the coefficients $a_{i,k}$'s can be learned based on 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 Fig. 4 to construct the calibration function for each sensor and then use all the data for testing. Fig. 4 also shows an example of calibrated CFD results with $K = 1$.

V. Real–Time Temperature Prediction

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

A. 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 paper, the period of data collection is referred to as *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 *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 high complexity of data center environment. In this work, we propose a linear prediction model to approximate $f_k(\cdot)$, which allows the online real–time prediction at low overhead. Suppose $\mathbf{p} = [p_1; p_2; \ldots]$, 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 the 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 the

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

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 Eq. (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, Eq. (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.

B. 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 the 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 shutdown during the CFD transient simulation. Suppose a data set with 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, we aim to find a matrix $\mathbf{A}_k$ to minimize $\sum_{t=R}^{L-k} \|\mathbf{t}(t+k)-\hat{\mathbf{t}}(t+k)\|_{\ell_2}^2 = \sum_{t=R}^{L-k} \|\mathbf{t}(t+k)-\mathbf{A}_k\cdot\mathbf{x}(t)\|_{\ell_2}^2$, where $\|\cdot\|_{\ell_2}$ represents the Euclidean norm. A desirable property of the above formulation is that the problem can be decomposed to the sub-problems 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 sub-problem is to find $\mathbf{a}_j$ to minimize $\sum_{t=R}^{L-k} (t_j(t+k) - \mathbf{a}_j \cdot \mathbf{x}(t))^2$. The closed-form solution of $\mathbf{a}_j$ is given by $\mathbf{a}_j = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{t}_j$, 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.

VI. System Implementation and Deployment

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

A. Testbeds and Sensor Deployment

Our first single-rack testbed, shown in Fig. 5(a), consists of a rack of 15 1U [2] servers in a 5×6 square feet 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 [25] is placed out of 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 of 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 [26] temperature sensors are mounted with brackets at the inlets of the 5 group of servers, and another 15 temperature sensors (8 Iris and 7 TelosB [26]) are mounted with brackets at the outlets of these servers. At the ceiling vent, a temperature sensor (TelosB) is mounted with bracket and a F333 airflow velocity sensor [27] 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 a same 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.

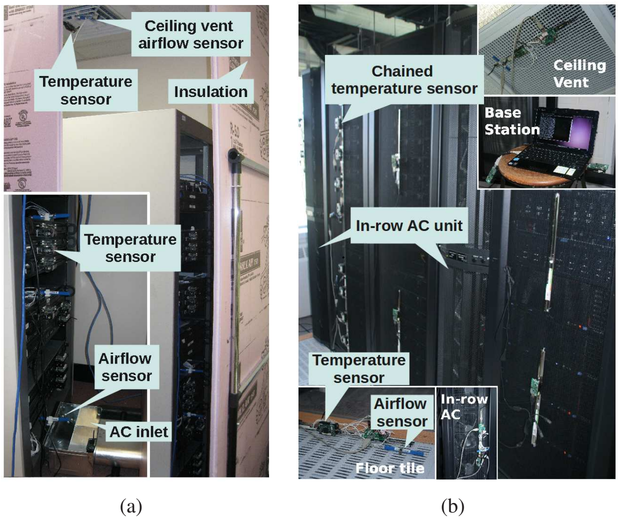


Figure 5. (a) Single-rack testbed; (b) Prodution testbed (HPCC)

Fig. 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 2016 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 Fig. 5(b). In addition to the raised floor cooling system which 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 the 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

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

to monitor the inlets and outlets temperatures, respectively. For one rack, we deploy 8 sensors evenly to monitor the server inlets and 8 sensors to monitor the server outlets, respectively. 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 sensor and airflow sensor 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 from Fig. 11.

B. Implementation of the Sensor Network

1) Wireless Sensors: In each of our testbed implementations, we use a single-hop network architecture where the base station sends the 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 multi-hop network topology can be employed when more server racks need to be monitored. As this collection scheme works in a time-division fashion, the system does not generate many collisions between the data transmissions of different sensors. TelosB [26] and Iris [26] 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 [27] via I2C interface. The programs on these motes are implemented in TinyOS 2.1 [28].

2) 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 the CPU utilization, and report fan speed from either `lm-sensors` or `ipmimonitoring` 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.

VII. Performance Evaluation

To evaluate the performance of our prediction system, we conduct extensive experiments on the single-rack testbed and the 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.

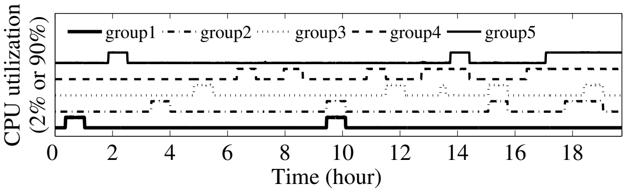


Figure 6. CPU utilization of the training data

A. Single-Rack Testbed Experiments

Fig. 2(a) shows the server groups and the temperature sensor locations on the rack of single-rack testbed. Five server groups, denoted as 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 [6]. 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.

1) Predication under Dynamic Workloads: The first experiment evaluates the performance of our system in response to the CPU utilization changes. A total of 25 hours of data were collected during 6 days. As the infrastructure ceiling vent is regularly shut down every night, we concatenate the data collected in different days when the ceiling 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 $k = 10\,\text{min}$. Fig. 6 shows CPU utilization of the training data and Fig. 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 Fig. 7, at about the 30th minute from the start, the temperature reached equilibrium. In the first 3 hours, as 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 dashed curve is calculated using measurements of all sensors 10 minutes before. While the prediction results well match the sensor measurements during the first 3 hours, however, we observe a considerable gap between the predicted temperatures and sensor measurements in 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. According to the multi-horizon prediction scheme discussed in Section VII-A2, the gap can be shortened by setting a

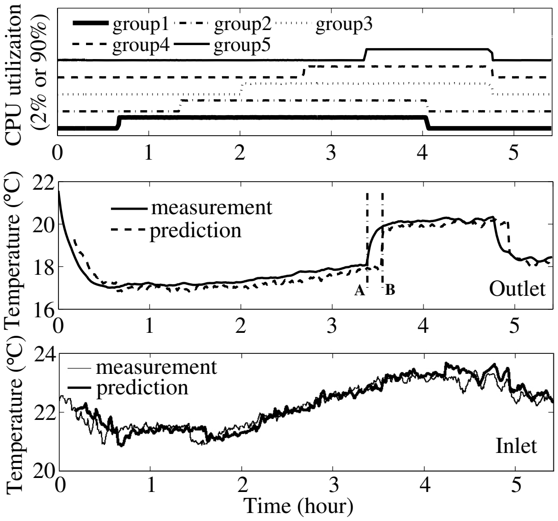


Figure 7. Top: CPU utilization of testing data. Middle: temperature measurements and predictions at an outlet of Group 5 with 10 minutes prediction horizon. Bottom: temperature measurements and predictions at an inlet of Group 3 with 10 minutes prediction horizon.

smaller prediction horizon. 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.

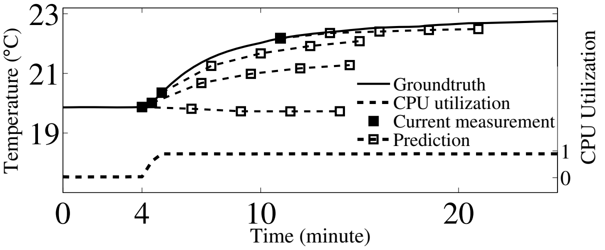


Figure 8. Temperature evolution prediction. Each solid rectangle represents the temperature measurement at current time instance and the white rectangles are the predicted temperatures at four different prediction horizons (0.5, 2.5, 5 and 7.5 minutes).

2) Multi-Horizon Prediction: In our prediction system, by training models with different k in Eq. (1), we can build multiple models to predict the evolution of temperature in the future. Fig. 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 started 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. Fig. 9 shows the root-mean-square error (RMSE) of multi-horizon predictions for each sensor. We can see that the RMSE generally increases with the prediction horizon. This conforms to the intuition that the temperature at a farther 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.

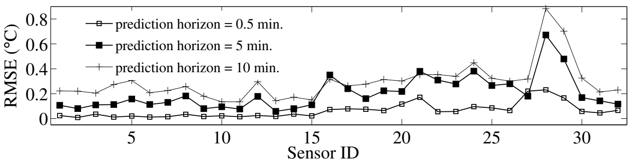


Figure 9. Root-mean-square error (RMSE) of multi-horizon temperature prediction.

3) Multi-Channel Prediction: In this experiment, we evaluate the accuracy of prediction in multiple thermal conditions (i.e., channels). As the 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 the AC failure, respectively. A total of 10 hours of data were collected when the servers run in 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 be 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.

Fig. 10 shows the absolute prediction errors of the two channels with respect to the groundtruth sensor measurements. The system exhibits very small absolute error in normal state, while it suffers up to 6°C absolute error during the AC failure. This is because the training data for the normal channel do not capture this abnormal situation. On the contrary, AC failure channel exhibits slightly higher absolute error than the normal channel during the normal state, while it has significantly lower absolute error during

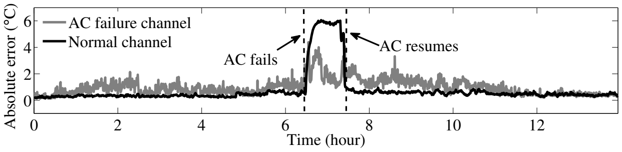


Figure 10. Average absolute temperature prediction error (prediction horizon = 5 minutes)

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 by existing data fusion techniques [29].

B. Production Testbed Experiments

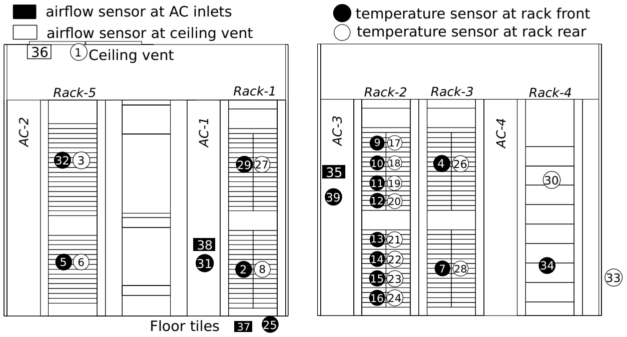


Figure 11. Front view of the two rows of racks, which face to each other in the server room

We also deployed and evaluated our system on a production testbed in a server room of High Performance Computer Center (HPCC) at Michigan State University. In this testbed, we deployed 35 temperature sensors and 4 airflow velocity sensors. Fig. 11 shows the sensor deployment from the front view of the two rack rows. We deploy 16 sensors on one rack (Rack-2) while other racks are instrumented with 2 or 4 sensors. Different from the single-rack testbed whose CPU utilization is controlled, the CPU utilization in HPCC testbed is very dynamic, which makes the accurate temperature prediction more challenging.

We evaluate our prediction approach in HPCC testbed using the data collected in 15 continuous days. The data from the first three days (March 31 to April 2, 2012) are used as training data, while the data of the following 12 days are used for prediction evaluation. Fig. 12 shows two examples of prediction results and the groundtruth temperature measurements. The prediction horizon is set to be 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. Fig. 13 shows the average absolute prediction error and the 90% error bound for all sensor locations. We can observe that the prediction errors on outlet sensors are slightly higher than 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 settings of prediction horizons. Fig. 14 shows the empirical Cumulative Distribution Function (CDF) of prediction error over all sensor locations. Similar to the results in Fig. 10, the prediction error increases with the prediction horizon.

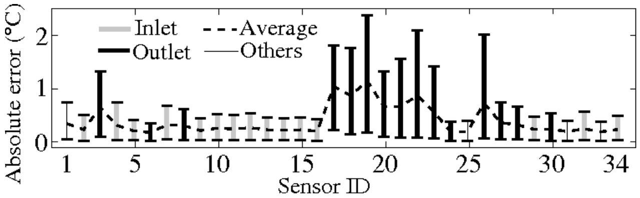


Figure 13. Absolute errors with the 90% error bound for each sensor with 10 minutes prediction horizon.

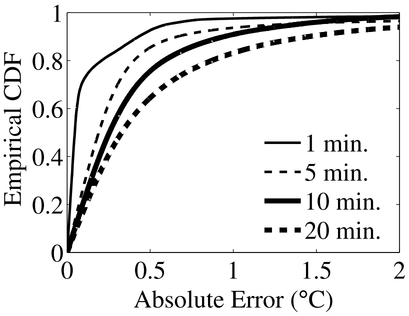


Figure 14. Empirical CDF of absolute error for all sensors with different prediction horizons.

VIII. Conclusion and Future Work

In this paper, we describe the design and implementation of a novel cyber-physical system for predicting temperature distribution of data centers. Our approach integrates Computational Fluid Dynamics (CFD) modeling and real-time data-driven prediction to achieve high fidelity temperature forecasting in various thermal conditions of data centers, including rare but critical thermal emergency situations like AC failures. We have implemented the system on a single-rack testbed and a testbed of 5 racks and 232 servers in a production high performance computing center. Extensive experimental results show that our approach can accurately

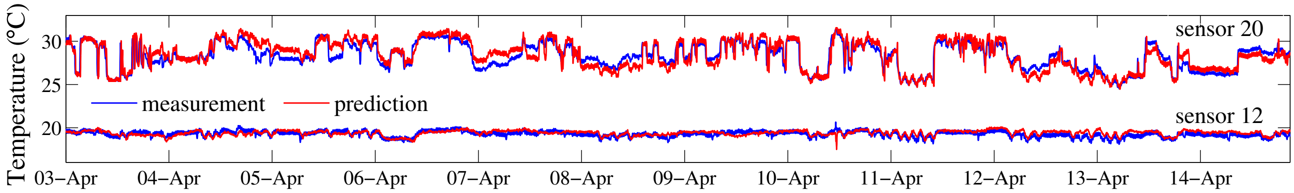


Figure 12. Long-term monitoring with 10 minutes prediction horizon. Sensor 20 and sensor 12 are located at server outlet and inlet, respectively.

predict the 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 the 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.

IX. Acknowledgment

This research was supported in part by the U.S. National Science Foundation under grants CNS-0954039 (CAREER), CNS-1218475 and CNS-1218154.

References

[1] Aperture Research Institute, "Data center professionals turn to high-density computing as major boom continues," *Research Note*, April 2007.

[2] http://blog.wikimedia.org/2010/03/24/ global-outage-cooling-failure-and-dns/.

[3] U.S. Environmental Protection Agency, "Report to congress on server and data center energy efficiency," 2007.

[4] C. Bash, C. Patel, and R. Sharma, "Dynamic thermal management of air cooled data centers," in *ITherm*, 2006.

[5] C. Bash and G. Forman, "Cool job alloction: Measuring the power savings of placing jobs at cooling-efficient locations in the data center," in *USENIX ATC*, 2007.

[6] J. Moore, J. Chasey, P. Ranganathanz, and R. Sharmaz, "Making scheduling"cool": Temperature-aware workload placement in data centers," in *USENIX ATC*, 2005.

[7] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," in *IEEE Trans. Parallel Distrib. Sys.*, 2008.

[8] J. Anderson and J. Wendt, *Computational fluid dynamics*. McGraw-Hill, 1995, vol. 206.

[9] T. Heath, A. P. Centeno, P. George, L. Ramos, Y. Jaluria, and R. Bianchini, "Mercury and freon: temperature emulation and management for server systems," in *ACM ASPLOS*, 2006.

[10] L. Ramos and R. Bianchini, "C-oracle: Predictive thermal management for data centers," in *HPCA*, 2008.

[11] Q. Tang, T. Mukherjee, S. K. S. Gupta, and P. Cayton, "Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters," in *ICISIP*, 2006.

[12] J. Moore, J. Chasey, and P. Ranganathanz, "Weatherman: Automated, online, and predictive thermal mapping and management," in *ICAC*, 2006.

[13] L. Li, C.-J. M. Liang, J. Liu, S. Nath, A. Terzis, and C. Faloutsos, "Thermocast: A cyber-physical forecasting model for data centers," in *ACM KDD*, 2011.

[14] J. Niemann, "Best practices for designing data centers with the infrastruxure inrow RC," *Application note of American Power Conversion*, 2006.

[15] G. C. Bell, "Improving data center efficiency with rack or row cooling devices: Results of "chill-off 2" comparative testing," *Federal Energy Management Program*, 2012.

[16] N. Rasmussen, "Cooling options for rack equipment with side-to-side airflow," http://www.chesapeakemc.com/APC_White_Papers/cooling_for_rack_equipment.pdf.

[17] C. J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao, "RACNet: A high-fidelity data center sensing network," in *ACM SenSys*, 2009.

[18] S. Choochaisri, V. Niennattrakul, S. Jenjaturong, C. Intanagonwiwat, and C. A. Ratanamahatana, "Senvm: Server environment monitoring and controlling system for small data center using wireless sensor network," in *International Computer Science and Engineering Conference*, 2010.

[19] X. Wang, X. Wang, G. Xing, J. Chen, C.-X. Lin, and Y. Chen, "Towards optimal sensor placement for hot server detection in data centers," in *ICDCS*, 2011.

[20] C. Mansley, J. Connell, C. Isci, J. Lenchner, J. O. Kephart, S. McIntosh, and M. Schappert, "Robotic mapping and monitoring of data centers," in *ICRA*, 2011.

[21] J. Lenchner, C. Isci, J. Kephart, C. Mansley, J. Connell, and S. McIntosh, "Toward data center self-diagnosis using a mobile robot," in *ICAC*, 2011.

[22] ASHRAE Techinical Committee 9.9, "2011 thermal guidelines for data processing environments expanded data center classes and usage guidance."

[23] Active Power, Inc., "Data center thermal runaway: A review of cooling challenges in high density mission critical environments," 2007.

[24] U. Singh, A. K. Singh, P. S, and A. Sivasubramaniam, "CFD-based operational thermal efficiency improvement of a production data center," in *1st USENIX Workshop on Sustainable Information Technology*, 2010.

[25] Tripp Lite, Inc., Portable AC Unit SRCOOL12K.

[26] Memsic Corp., "Telosb, Iris datasheets, 2012."

[27] Degree Controls, Inc., "F333 airflow sensor user guide."

[28] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An operating system for sensor networks," *Ambient Intelligence*, vol. 35, 2005.

[29] P. K. Varshney, *Distributed Detection and Data Fusion*. Springer, 1996.