PTEC: A System for Predictive Thermal and Energy Control in Data Centers

Jinzhu Chen* Rui Tan‡ Guoliang Xing* Xiaorui Wang†
*Michigan State University, USA †Ohio State University, USA
‡Advanced Digital Sciences Center, Illinois at Singapore

***Abstract*—Current data centers often adopt conservative and static settings for cooling and air circulation systems, leading to excessive energy consumption. This paper presents the design and evaluation of PTEC – a system for predictive thermal and energy control in data centers. PTEC leverages the server built-in sensors and monitoring utilities, as well as a wireless sensor network, to monitor both the cyber and physical status of a data center. By predicting the temperature evolution of a data center in real time, PTEC finds the temperature setpoints, the cold air supply rates, and the speeds of server internal fans to minimize the expected total energy consumption of cooling and circulation systems. Moreover, PTEC enforces the upper bounds on server inlet temperatures and their temporal variations to prevent server overheating and reduce server hardware failure rate. We evaluated PTEC on a hardware testbed consisting of 15 servers and a total of 23 temperature and power sensors, as well as through Computational Fluid Dynamics (CFD) simulations based on real data traces collected from a data center with 229 servers. The experimental results show that PTEC can reduce the cooling and circulation energy consumption by more than 30%, compared with baseline thermal control strategies.**

I. Introduction

Thermal and energy management has become a key challenge in the design and operation of data centers. A recent worldwide data center survey shows that the non-computing energy takes average 45% and up to 60% of total energy [4]. One of the key reasons for these data centers to have excessive energy consumption is the inefficient operation of Computer Room Air Conditioning (CRAC) systems. Because of the lack of visibility in the operating conditions, the CRAC systems often use unnecessarily low temperature setpoints to reduce the risk of server overheating. Due to such a conservative strategy, the CRAC systems can account for up to half of the energy consumption of a data center [23]. Moreover, data centers usually maintain unnecessarily high levels of air circulation by adopting static settings or simplistic control strategies for the circulation systems including server fans. As a result, the server fans can take up to 23% of server power consumption [25]. Thus, improving the efficiency of cooling and circulation systems plays an important role in reducing the total energy consumption of a data center.

Various efforts have been made to improve data center energy efficiency. New green data center technologies have proven their effectiveness in a few latest industrial scale data centers. For instance, the new Google data centers reduce the non-computing energy ratios down to about 10% [1]. However, these technologies require a clean slate redesign and hence are cost prohibitive to apply in existing data centers. A recent survey reveals that 85% of existing data centers have non-computing energy ratios higher than 40% [4]. Therefore, low-cost effective thermal management systems that can retrofit existing data centers with better energy efficiency are highly appealing. Data center operational guidelines have been revised recently to avoid overly conservative settings. For instance, the American Society of Heating, Refrigerating and Air Conditioning Engineers (ASHRAE) recommends to increase the environmental temperatures in data centers up to 27°C to reduce cooling energy consumption [2]. However, without a real-time thermal control system that ensures the thermal safety, the higher environmental temperature setpoints will increase the risk of server overheating. In fact, a survey in 2013 shows that 90% of data centers still operate under 24°C [5].

A variety of thermal control schemes have been recently proposed to prevent thermal emergencies in a data center while reducing the energy costs. The existing approaches either optimize a single thermal variable (e.g., server workload, CRAC setpoint, or fan speed, etc.) [7] [19] or a combination of them [6] [14] to minimize the energy costs. However, most of these approaches are based on a *reactive* scheme, which reactively controls the cooling systems to eliminate detected hot spots. Unfortunately, this approach often cannot achieve desirable energy efficiency, primarily due to the complex thermodynamics of data centers. For instance, the heat generated by increased server workload takes substantial delays to be recirculated to the server inlets. To react to the detected hot spots at the server inlets, the CRAC systems need to adopt sufficiently low temperature setpoints, which, however, significantly downgrade their energy efficiency [19].

In contrast to the existing reactive thermal control schemes, this paper proposes a *proactive* approach to prevent potential future hot spots. Specifically, our approach predicts the energy consumption and thermal conditions resulted by each possible CRAC/circulation control action in real time, and then executes the best one. We need to address two major challenges to realize such a proactive control scheme. First, the thermal characteristics of a data center are inherently affected by both physical (e.g., complex airflows) and cyber (e.g., dynamic server workloads) factors. In particular, the temperature evolution, the energy consumption of CRAC/fan systems, and their control decisions are tightly coupled together. Moreover, the control decisions not only need to improve the energy efficiency, but also must account for servers' thermal safety requirements. Second, a large number of variables in a data center may affect the temperatures and the total energy consumption, including the fan speed of each server and the temperature setpoint of each CRAC system. The global energy optimization based on all controllable variables often has prohibitive computational complexity. Moreover, to ensure system reliability even in certain thermal emergencies, the thermal control system should not resort to the computing resources of the monitored data center.

This paper presents a real-time Predictive Thermal and Energy

Control (PTEC) system that improves the energy efficiency of both the cooling and circulation systems of a data center, while meeting a set of thermal safety requirements. PTEC leverages the server built-in sensors and monitoring utilities, as well as an external easy-to-deploy wireless sensor network to monitor both the cyber and physical status of a data center, which includes CPU utilization, dynamic air flow, temperature distribution, and CRAC/fan settings. Based on these measurements, PTEC predicts the server inlet temperatures in real time and proactively controls the temperature setpoints and blower speeds of CRAC systems, and the speeds of server fans, to reduce their overall energy consumption. PTEC enforces a set of thermal safety requirements including the upper bounds on server inlet temperatures and their temporal variations. A high inlet temperature directly indicates server overheating and potential server shutdowns, while a high temporal variation of server temperature can significantly increase hardware failure rate [12]. To make the energy optimization problem tractable, PTEC adopts a *coordinated control* approach. Specifically, a novel dynamic fan speed control algorithm is first developed to automatically control the server fans based on the server CPU utilization and the inlet temperature. Then, PTEC searches for the temperature setpoints and blower speeds of CRAC systems to minimize the overall energy consumption of CRAC systems and server fans. Moreover, we propose a partition-based algorithm, which divides the CRAC systems to multiple regions based on their spatial thermal correlations with the servers, to reduce the computational overhead of PTEC for large-scale data centers.

We prototyped PTEC and deployed it on a hardware testbed consisting of 15 servers and a total of 23 temperature and power sensors. The results show that PTEC can reduce the cooling and circulation energy consumption by up to 34% and 30%, compared with an overcooling strategy and a reactive control strategy, respectively. We also conducted trace-driven Computational Fluid Dynamics (CFD) simulations for an existing data center with 229 servers to validate the effectiveness and scalability of PTEC.

II. Related Work

Existing data center thermal control approaches can be broadly divided into two categories. The first category of approaches minimizes the energy consumption or a cost function by controlling a single type of thermal variables, e.g., server workloads [19], CRAC setpoints [7], fan speeds [25], CPU frequencies [17], or the number of virtual machines [16]. Thermal-aware load balancing (e.g., [19]) can prevent hotspots and help increase room temperature for energy saving. In [7], CRAC systems are controlled to maintain the temperatures of selected locations at their setpoints. In [25], fan speeds of blade servers are controlled to minimize the total fan power consumption subject to an upper bound on CPU temperatures. In [17], the sum of CPU frequencies is maximized under a certain power budget. All the above approaches focus on controlling one type of thermal variables to reduce the power consumption.

The second category of approaches controls multiple types of thermal variables to reduce the overall power consumption. In [22], server fan speeds, CPU power states and workload migration are controlled to minimize the overall power consumption within a blade server enclosure. In [21], server thermal management policies are selected based on predicted temperatures. However, these two approaches [21] [22] do not consider the cooling energy consumption. In [6], computing jobs are assigned to the servers with the highest cooling efficiencies. The CRAC setpoints are then calculated based on the job assignment to ensure thermal safety for servers. The approach in [20] minimizes the cost of electricity and quality of service degradation while maintaining the server temperature within a predefined range. In [14], the CRAC setpoints are first determined based on data center utilization levels, and then a feedback server fan control approach is applied to achieve a trade-off between server leakage power and fan power. However, it does not minimize the overall energy consumption of CRAC systems and server fans. The approach in [18] minimizes a holistic cost metric, accounting for the availabilities and prices of the traditional/renewable electricity sources, multiple cooling options, and data center workload status. However, this approach accounts for neither the variability of fan power, nor the complex dynamics of server temperatures. In [27], floor-mounted adaptive vent tiles and CRAC cooling provision are controlled to reduce the cooling cost. However, the server fan power consumption is not considered. Moreover, their approach cannot be readily applied to other cooling structures such as in-row cooling that is commonly adopted in data centers.

III. Problem Statement and Approach Overview

A. Problem Statement

The CRAC systems are the major source of energy consumption in many existing data centers [23]. Another thermal-related source of energy consumption is the server fans, which can take up to 23% of server power [25]. It has been shown that the cooling efficiency of a CRAC system increases with its temperature setpoint [19]. With a higher setpoint, a CRAC system can remove the same amount of heat with up to 40% less energy consumption [19]. However, a higher CRAC setpoint increases the likelihood of server overheating and heat-induced server shutdown. Moreover, it may adversely increase server fan speeds for removing the heat generated by the servers, resulting in higher overall energy consumption. This paper aims to design a control system to reduce the overall energy consumption of CRAC systems and server fans, subject to a set of thermal requirements including upper bounds on server inlet temperatures and their temporal variations. Upper-bounding them can prevent server overheating and severe temperature fluctuations that can cause high hardware failure rates [12].

A CRAC and server fan control system has to address the following two fundamental issues. First, the data center has complex thermodynamics and tight thermal coupling among the servers, the CRAC systems, and the physical environment. It is challenging to accurately model the thermodynamics, which, however, is the base for designing an effective thermal control system. Second, to adapt to the unpredictable dynamics of the data center, the control system must run in a real-time manner. However, the optimal control algorithm is computation-intensive due to the complex and non-linear relationships among control actions, energy efficiency, and thermal conditions.

This paper designs a data center thermal and energy control system based on a *predictive* scheme. This design choice is

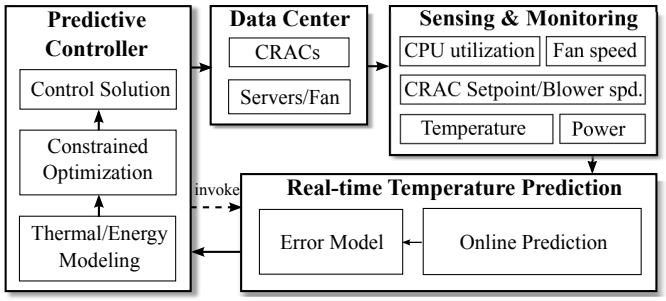

Fig. 1. PTEC system architecture.

based on the key observation that there are various time delays in the data center thermodynamics. For instance, the extra heat generated by suddenly increased server workload takes a considerable delay to be recirculated by the server fans and the CRAC systems to the server inlets. Moreover, due to limited cooling capacity, it typically takes a considerable delay for a CRAC system to reach the new temperature setpoint. Thus, it is desirable to *proactively* control the CRAC/fan systems in an energy-efficient manner to prevent potential future hot spots.

B. Approach Overview

Fig. 1 illustrates the architecture of PTEC based on a predictive control scheme. It consists of three major components:

Sensing and monitoring: PTEC periodically collects the measurements of the variables that affect the temperatures in a data center, which include server status (CPU utilizations, system temperatures, fan speeds, and powers) and CRAC status (powers, temperature setpoints, and blower speeds) from the built-in sensors and monitoring utilities, and a few critical temperatures (e.g., server inlet and CRAC return hot air temperatures) from a small number of wireless sensors. These wireless sensors, powered by either onboard batteries or USB interface of servers, can self-organize into a network for data collection. Therefore, the overhead of sensor installation process is small. Fig. 9 shows the sensor deployment on a single rack testbed.

Real-time temperature prediction: The system can rapidly predict the evolution of temperature distribution based on the collected sensor data and a candidate CRAC/fan control solution. Such a real-time prediction enables the system to assess a large number of candidate control solutions during runtime.

Predictive controller: We model the power consumption of server fans and CRAC systems. Based on the models, we formally formulate the problem of minimizing the predicted overall energy consumption of server fans and CRAC systems, subject to a set of thermal safety requirements. A predictive controller assesses the temperature evolution in the future for each candidate control solution and chooses the most energy-efficient one. The solution comprises the temperature setpoints, blower speeds of CRAC systems, and server fan speeds.

IV. Power Consumption Models

A. Server Fan Power Consumption Model

Air circulation is critical for cooling servers in a data center. A server system is typically equipped with several internal fans for cooling different components, such as power supply unit and

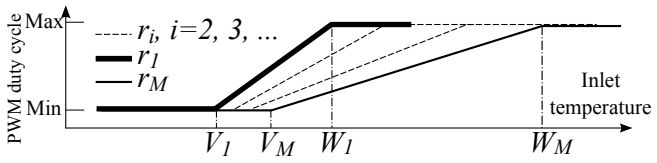

Fig. 2. PWM duty cycle vs. server inlet temperature in fan speed control. Curve r_1 is the native fan speed control algorithm. Our new Dynamic Fan Speed Control algorithm consists of the curves $r_1, r_2, \ldots, r_M$ (cf. Section V-C).

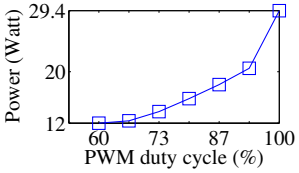

Fig. 3. Power of a server fan vs. PWM duty cycle.

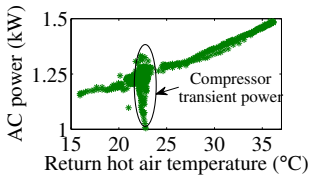

Fig. 4. AC power vs. return hot air temperature.

CPU. A server fan often regulates its speed according to the duty cycle of a pulse width modulation (PWM) signal. Most servers control the fan speed using a simple native algorithm, which linearly increases the PWM duty cycle based on the server inlet temperature [3]. In Fig. 2, curve r_1 illustrates the input and output of the algorithm, which is parameterized by two thresholds V_1 and W_1. When the server inlet temperature is lower than V_1, the algorithm sets a minimal PWM duty cycle to maintain the lowest allowed fan speed. When the inlet temperature exceeds V_1, the algorithm increases the PWM duty cycle (and hence the fan speed) linearly with the inlet temperature. When the inlet temperature exceeds W_1, the algorithm sets the maximal PWM duty cycle, maintaining the full fan speed.

The relationship between the fan power and the PWM duty cycle can be estimated from either off-line experiments or on-line measurements. We conduct an experiment to measure the power of a server fan (Delta Electronics BFB1012EH) under various PMW duty cycles. The results are shown in Fig. 3. Let $D(T)$ denote the PWM duty cycle determined by the native control algorithm, where T is the inlet temperature. We adopt a discrete-time model with equal time steps. Let t_n denote the time instant of the n-th time step. For the l-th server, let $P_{Fl}(D(T_l(t_n))$ denote the instantaneous power of its fan at time instant t_n, where T_l is the inlet temperature. Therefore, with the predicted inlet temperature $T_l(t_{n+k})$ at time instant t_{n+k}, the predicted server fan instantaneous power is given by $P_{Fl}(D(T_l(t_{n+k}))$, which is abbreviated as $P_{Fl}(t_n, k)$ hereafter.

B. CRAC Power Consumption Model

The power of a CRAC system is mainly determined by its temperature setpoint, blower speed, and return hot air temperature. At time instant t_n, the power consumption of the j-th CRAC system is denoted as $P_{Cj}(S_j(t_n), B_j(t_n), T_{Hj}(t_n))$, where $S_j(t_n)$, $B_j(t_n)$, and $T_{Hj}(t_n)$ represent the temperature setpoint, the blower speed, and the return hot air temperature for this CRAC system. For a CRAC system with continuous setpoint and blower speed, the parameters of P_{Cj} can be obtained by interpolation based on off-line measurements. With the predicted return hot air temperature $T_{Hj}(t_{n+k})$ at time instant t_{n+k}, the predicted instantaneous power of the j-th CRAC system is given

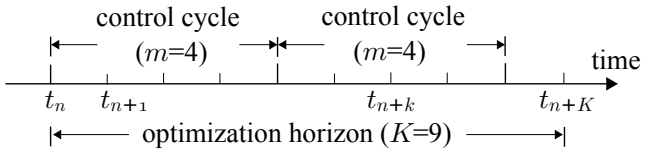

Fig. 5. Predictive control scheme.

by $P_{Cj}(S_j(t_{n+k}), B_j(t_{n+k}), T_{Hj}(t_{n+k}))$, which is abbreviated as $P_{Cj}(t_n, k)$ hereafter.

As an example, we empirically study the power consumption model of a Tripp Lite SRCOOL12K air conditioner (AC), which is specially designed for data centers. The resulted model will be also used in our testbed experiments in Section VII. This AC has three selectable blower speeds and a maximum cooling power of 1.5 kW. Fig. 4 shows that the power of this AC almost linearly increases with the return hot air temperature when the compressor is not in a transient state. When the compressor is off, its power is almost constant (not shown in Fig. 4) if the blower speed is fixed. Moreover, as different blower speeds result in less than 10 W power difference regardless of the compressor status, we round up the power consumption of the blower to its maximal value to simplify the model. When the compressor transits from off to on, there is a spike of transient power. However, this spike lasts for about 15 seconds only and takes about 0.5% of the total energy under normal settings. Therefore, it is neglected in our power consumption model. This AC does not allow us to program the temperature setpoint. Therefore, for this particular AC, we only control the on/off states of the blower and the compressor. As a result, the status of this AC can be represented by two binary variables, which are the compressor status $S \in \{0, 1\}$ and the blower status $B \in \{0, 1\}$, where 0 and 1 represent off and on states. The instantaneous power of this AC can be described by $P_C = B \cdot [S \cdot (\omega_1 T_H + \omega_0) + \omega_2]$, where the parameters ω_0, ω_1, and ω_2 can be estimated from the data shown in Fig. 4.

V. Design of PTEC

A. Problem Formulation

PTEC adopts a predictive control scheme illustrated in Fig. 5. Suppose the current time instant is t_n. The time interval between t_n and t_{n+1} is referred to as a *time step*, which is the period for sensor sampling and temperature prediction. Our system collects measurements from sensors at the beginning of every time step. A *control cycle* is defined as m consecutive time steps. At the beginning of each control cycle, PTEC determines the CRAC settings and the server fan speeds to minimize the predicted overall power consumption of the CRAC systems and server fans subject to a set of thermal requirements during the following K time steps (i.e., from t_n to t_{n+K}), where K is the *optimization horizon*. Based on this scheme, this section formulates the predictive control problem.

For a total of J CRAC systems and L servers, the predicted average power consumption during the future K time steps is

$$\overline{P(t_n)} = \frac{1}{K} \sum_{k=1}^{K} \left(\sum_{j=1}^{J} P_{Cj}(t_n, k) + \sum_{l=1}^{L} P_{Fl}(t_n, k) \right), \quad (1)$$

where P_{Cj} is the power consumption of the j-th CRAC system and P_{Fl} is the fan power consumption of the l-th server. We note that the server temperatures also affect the leakage powers of server electronics. However, their temperature-induced changes are negligible compared to the power consumption of server fans [12]. Therefore, our objective function in Eq. (1) does not account for server leakage power.

PTEC enforces that the servers will not be overheated in the future K time steps. Let $T_{l,k}$ and T_U denote the inlet temperature of the l-th server at time instant t_{n+k} and the *maximum allowed temperature* (MAT) at any server inlet, respectively. PTEC aims to ensure $T_{l,k} < T_U$, $\forall l \in [1, L]$ and $\forall k \in [1, K]$. A challenge in the design of any predictive control system is how to cope with prediction errors [10] [15]. Let $\hat{T}_{l,k}$ denote the predicted inlet temperature for the l-th server at the prediction horizon k. We assume that the prediction error (i.e., $\hat{T}_{l,k} - T_{l,k}$) follows the normal distribution $\mathcal{N}(\mu_{l,k}, \sigma^2_{l,k})$, which will be empirically estimated in Section V-B. Thus, the actual temperature $T_{l,k} \sim \mathcal{N}(\hat{T}_{l,k} - \mu_{l,k}, \sigma^2_{l,k})$. PTEC requires that the actual temperature $T_{l,k}$ is lower than T_U with a confidence level α, i.e., $\Pr(T_{l,k} < T_U) > \alpha$. Therefore, the upper bound for $\hat{T}_{l,k}$, denoted by $\widetilde{T}_{l,k}$, can be derived as

$$\widetilde{T}_{l,k} = T_U + \mu_{l,k} - \sigma_{l,k} Q^{-1}(1 - \alpha), \quad (2)$$

where $Q(\cdot)$ is the Q-function of the standard normal distribution, i.e., $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} \exp(-\frac{z^2}{2}) \mathrm{d}z$. The parameters $\mu_{l,k}$ and $\sigma^2_{l,k}$ should be estimated for each server l and prediction horizon k. They can be continuously updated using the most recent historical measurements. Note that the above approach can be extended to address other random distributions for the prediction error, given the distribution of $T_{l,k}$ that is derived from $\hat{T}_{l,k}$ and the prediction error's distribution.

A data center should be prevented from significant temperature variation. The probability of a server outage can be doubled when the temporal temperature variation increases by 50% [12]. PTEC computes the temperature variation over a moving window with size of $w \geq K$, from time instant $t_{n-(w-K)}$ to t_{n+K}. The values before the time instant t_n are historical measurements and the values after that are predicted temperatures. We use the *relative standard deviation* (RSD) to quantify the temperature variation, i.e., $\mathrm{RSD} = \sigma_T / \mu_T$, where σ_T and μ_T are the standard deviation and mean of the temperatures in the moving window. We denote RSD_l the RSD for the l-th server. PTEC requires that the RSD of each server is upper-bounded by a constant RSD_U specified by the data center operator. For instance, the setting $\mathrm{RSD}_U = 0.04$ can maintain a satisfactorily low fluctuation-induced hardware error rate [12].

We now formally formulate the thermal and energy control problem that minimizes the power consumption of CRACs and server fans. For a total of J CRACs and L servers, let $\mathbf{D}$, $\mathbf{S}$, and $\mathbf{B}$ denote the vectors of server fan PWM duty cycles, CRAC temperature setpoints and blower speeds over the optimization horizon, respectively, i.e., $\mathbf{D} = [D_1, \ldots, D_L]$, $\mathbf{S} = [S_1, \ldots, S_J]$, and $\mathbf{B} = [B_1, \ldots, B_J]$. We formulate the problem as follows:

Problem 1: To find $\mathbf{D}$, $\mathbf{S}$, and $\mathbf{B}$ to minimize the predicted average power consumption given by Eq. (1), subject to that, $\forall l \in [1, L]$, $\forall k \in [1, K]$,

1) the predicted inlet temperatures are lower than an upper bound: $\hat{T}_{l,k} \leq \widetilde{T}_{l,k}$, where $\widetilde{T}_{l,k}$ is given by Eq. (2); and

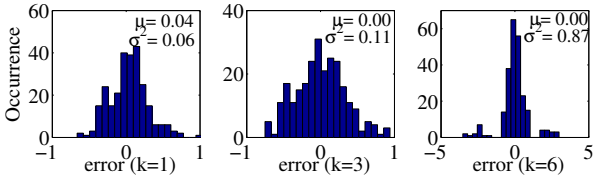

Fig. 6. Histograms of prediction errors with k = 1, 3, and 6. (time step = 30s)

2) the RSDs of the inlet temperature are lower than an upper bound: $\text{RSD}_l \leq \text{RSD}_U$.

PTEC solves Problem 1 at the beginning of each control cycle and controls the CRAC systems and server fans according to the solution. Problem 1 is a non-linear constrained optimization problem with prohibitive computational complexity due to the complex thermal interactions between CRAC systems and server fans. In particular, the exhaustive search has an exponential complexity with respect to the total number of CRAC systems and server fans. To make the problem tractable and achieve satisfactory real-time performance, we propose a *coordinated control* approach. Specifically, the servers will control the fan speeds autonomously according to server inlet temperatures and CPU utilizations, by using an algorithm in Section V-C. Thus, the variables of Problem 1 are reduced to $\mathbf{S}$ and $\mathbf{B}$ only. It is important to note that, when PTEC assesses each candidate solution $\langle \mathbf{S}, \mathbf{B} \rangle$, the resulted fan speeds under the autonomous control algorithm will be used to calculate the predicted average power consumption, inlet temperatures, and RSDs. Thus, this coordinated control approach accounts for the interdependence between server fans and CRAC systems. Therefore, it does not substantially degrade the solution quality.

B. Real-Time Temperature Prediction

PTEC integrates a real-time data-driven temperature prediction algorithm [10] that predicts server inlet temperatures based on cyber and physical status of the data center. This section briefly reviews the algorithm. The input of the algorithm includes the temperatures at a set of selected locations, e.g., server inlets and CRAC hot air return registers, measured by either a deployed wireless sensor network or server built-in sensors. For a total of D temperatures, we define the temperature distribution $\mathbf{T} = [T_1; T_2; \ldots; T_D] \in \mathbb{R}^{D \times 1}$, where T_d is the temperature at the d-th location in the data center. The temperature distribution is predicted by a set of *thermal variable*s that significantly affect $\mathbf{T}$ and are monitored by sensors. They include the CRAC setpoints $\mathbf{S}$ and blower speeds $\mathbf{B}$, server fan speed control settings $\mathbf{R}$, and CPU utilizations $\mathbf{U}$. Moreover, the historical temperature distributions also largely affect the temperature distributions in the near future. Therefore, we define the *state* of the monitored data center at a time instant, denoted by $\mathbf{p}$, as a collection of thermal variables, i.e., $\mathbf{p} = [\mathbf{S}; \mathbf{B}; \mathbf{R}; \mathbf{T}; \mathbf{U}]$. The state $\mathbf{p}$ is measured every time step. At the beginning of a control cycle, the temperature distribution at time instant t_{n+k}, denoted by $\mathbf{T}(t_{n+k})$, is predicted based on the most recent R states. By setting increasing prediction horizon k, the algorithm predicts the temporal evolution of $\mathbf{T}$. This machine-learning-based prediction algorithm, which is trained with data from real sensors and off-line CFD simulations, achieves a desirable trade-off between prediction fidelity and computation overhead. It is

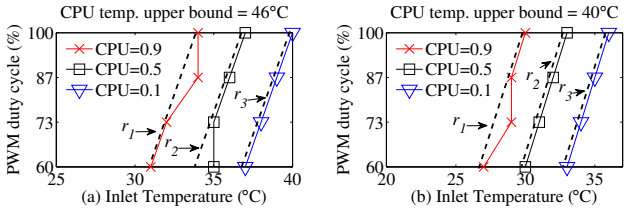

Fig. 7. Minimal required PWM duty cycle (marked curve) vs. server inlet temperature under various CPU utilization. Sub-figures (a) and (b) are the results for different CPU temperature upper bounds (46°C and 40°C). A DFSC setting r_i comprises two endpoints of a dashed line.

shown that the average prediction errors are less than 1°C when the prediction horizon $k = 10$ minutes for a production data center [10]. Moreover, its high-speed feature allows PTEC to iteratively search for the best control solution (cf. Section V-D).

Our temperature prediction approach assumes a fixed CPU utilization during the prediction horizon. By integrating existing server workload prediction algorithms (e.g., [13]), the accuracy of temperature prediction can be improved under dynamic workload. For instance, if the server workload is predicted to change significantly at time instant $t_{n+K'}$, where $0 < K' < K$, the temperature distributions $\mathbf{T}(t_{n+k})$ where $K' \leq k \leq K$ can be predicted based on the state $\mathbf{p}_{n+K'}$ that incorporates predicted workload.

Fig. 6 shows the histograms of temperature prediction errors for a server inlet. The errors can be well characterized by normal distributions. Consistent with intuition, the error variance increases with the prediction horizon. The error means and variances for different prediction horizons are used in Eq. (2).

C. Dynamic Fan Speed Control

The main purpose of server fan is to prevent the internal electronic components, e.g., CPU, from overheating. A key drawback of the native fan speed control approach discussed in Section IV-A is the neglect of the server status (e.g., CPU utilization) that also affects component temperatures. For instance, the fan may run at an unnecessarily high speed when the server is idle. Several existing fan control algorithms [14] [26] account for the CPU workload. However, it is often difficult to predict the fan power consumption when the system is operated under these algorithms. Thus, they cannot be readily integrated with PTEC that aims to minimize the total power consumption of CRAC systems and server fans.

Our new fan speed control approach, called Dynamic Fan Speed Control (DFSC), jointly considers CPU utilization and inlet temperature. Fig. 7 shows the minimal fan speeds (in PWM duty cycle) to meet two given upper bounds of CPU temperature (46°C and 40°C) versus server inlet temperature, under various CPU utilizations. We can observe that the minimal fan speed has a near-linear relationship with the inlet temperature. More importantly, such a relationship varies with the CPU utilization. Thus, DFSC reuses the native fan speed control algorithm but adjusts its setting in response to the CPU utilization changes while meeting a CPU temperature upper bound requirement. Specifically, DFSC discretizes the CPU utilization into M levels. The first level represents full utilization while the M-th level represents idle. Each level is mapped to a setting of two thresholds of the native control algorithm. As illustrated

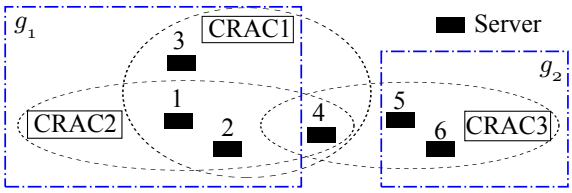

Fig. 8. Example of partitioning. The servers within an oval are associated with the CRAC in the oval. Region g_2 contains CRAC3 only since Server5 and Server6 are associated with CRAC3 only. CRAC1 forms a region since Server3 is associated with CRAC1 only. No servers are associated with CRAC2 exclusively. Therefore, CRAC2 will be merged with CRAC1 to form region g_1.

in Fig. 2, the setting for the i-th CPU utilization level is denoted by $r_i = \langle V_i, W_i \rangle$, where $i \in [1, M]$. When the CPU utilization is at the i-th level, the native algorithm will be invoked with the setting r_i. Under the setting r_1, the CPU temperature upper bound should be met when the server is fully utilized. Similarly, under the setting r_M, the CPU temperature upper bound should be met when the server is idle. The settings $\{r_1, r_2, \ldots, r_M\}$ are hardware-dependent and can be empirically measured, e.g., through off-line experiments, or provided by hardware manufacturers. The servers can also run an on-line feedback fan controller [14] for a certain period of time to measure these settings when the CPU temperature is stable. Once the settings are measured, the servers can start using DFSC. In Fig. 7, the endpoints of dashed lines show the DFSC settings with $M = 3$.

D. Predictive Controller

In our current implementation of PTEC, we use the constrained simulated annealing (CSA) algorithm [24] to search for the CRAC setting ($\mathbf{S}$ and $\mathbf{B}$) that minimizes Eq. (1) subject to the thermal safety requirements. The CSA algorithm is more efficient than the brute-force search and can asymptotically converge to the optimal solution [24]. For instance, for 229 servers and 4 CRAC units, each of which has 6 different states of setpoint and blower speed, it takes the CSA algorithm only 5 seconds on a 3.4GHz desktop computer to converge to a near-optimal solution with an optimization horizon of 8.

We now discuss the settings of control cycle m and optimization horizon K. First, intuitively, the system with a short control cycle can respond to the thermal condition changes quickly. However, a short control cycle allows less time for solving the optimization problem. Second, it is desirable to set a larger optimization horizon K such that the predictive controller accounts for a longer period of thermal dynamics into the future. However, its setting must also consider both the prediction accuracy and the controller's computational overhead. Therefore, the settings for m and K should achieve a satisfactory trade-off between control quality and computational overhead. In our testbed experiment with 6 servers and a portable AC, the control decision can be computed within a second when K is set to 6 to 9 minutes. Therefore, we set m = 1 to 3 minutes since the AC should not be switched frequently. Under these settings, nearly 90% of prediction errors are within 1°C.

E. Scalable Partition-Based Predictive Controller

Due to the non-convexity of Problem 1, the computational overhead of CSA increases exponentially with the number of CRAC systems. The resulted delay may jeopardize the real-time performance of PTEC for large-scale data centers. This section presents a partition-based algorithm that can significantly reduce the computational overhead while maintaining satisfactory solution quality. It consists of an *off-line stage* and an *on-line stage*. The off-line stage partitions the data center into several regions based on the thermal correlation index (TCI) [7], which characterizes the cooling effectiveness for a location provided by a CRAC system. The on-line stage solves Problem 1 within each region, and iteratively revises the sub-solution for each region to meet the thermal requirements of the servers out of any regions.

The TCI for the l-th server and the j-th CRAC system is defined as $\gamma_{l,j} = \frac{\Delta T_l}{\Delta T_{Cj}}$, where ΔT_{Cj} denotes a step change of the supply air temperature of the j-th CRAC system and ΔT_l is the resulted steady temperature change at the l-th server inlet. A larger $\gamma_{l,j}$ indicates a stronger capability of the CRAC system to remove the heat from the server. TCI can be experimentally measured by perturbing CRAC setpoints or numerically obtained from Computational Fluid Dynamics simuations [7]. The l-th server and the j-th CRAC system are *associated* if $\gamma_{l,j} \geq \lambda$, where $\lambda \in (0, 1)$ is a threshold specified by the system operator. Thus, the temperature at the l-th server's inlet is mainly affected by its associated CRAC systems. For instance, in Fig. 8, Server5 and Server6 are associated with CRAC3 only, and their inlet temperatures are mainly affected by CRAC3.

The off-line stage partitions the data center into a number of regions based on the TCIs. For a region denoted by g, let C_g denote a set of CRAC systems in this region and E_g denote a set of servers that are associated with the CRAC systems in C_g *exclusively*. Formally, $E_g = \{l | \gamma_{l,j} \geq \lambda, \exists j \in C_g, \forall l\} \cap \{l | \gamma_{l,j} < \lambda, \forall j \notin C_g, \forall l\}$. For example, the two dash-dotted rectangles in Fig. 8 show two such regions, denoted as g_1 and g_2. In this example, $C_{g_1} = \{\text{CRAC1}, \text{CRAC2}\}$, $C_{g_2} = \{\text{CRAC3}\}$, $E_{g_1} = \{\text{Server1}, \text{Server2}, \text{Server3}\}$, $E_{g_2} = \{\text{Sever5}, \text{Server6}\}$. Let C and $\mathbb{G}$ denote the set of all CRAC systems and the set of all regions. The off-line stage looks for a partition scheme to minimize $\max_{\forall g \in \mathbb{G}} |C_g|$ subject to 1) $\bigcup_{g \in \mathbb{G}} C_g = C$, 2) $C_g \cap C_h = \emptyset$, $\forall g \neq h$, and 3) $E_g \neq \emptyset$, $\forall g \in \mathbb{G}$. As we will solve Problem 1 within each region g in the on-line stage, the objective of minimizing the maximum number of CRAC systems in any region significantly reduces the computation overhead of the on-line stage. We develop a heuristic algorithm based on an existing independent set algorithm [9] to solve this partitioning problem. The details of the algorithm are omitted due to space constraints and can be found in [11]. We note that, some servers that are associated with many CRAC systems may be out of any E_g, e.g., Server4 in Fig. 8. These servers are called *ungrouped* servers.

Based on the regions partitioned by the off-line stage, the on-line stage solves Problem 1 as follows. Initially, for each region g, Problem 1 is solved for C_g and E_g using the CSA algorithm. The solution for a region is called a sub-solution. The initial solution for the data center thus comprises all sub-solutions. However, this solution may not meet the thermal requirements for the ungrouped servers. The on-line stage iteratively updates the solution by solving Problem 1 for each region plus all these ungrouped servers. As an ungrouped server is cooled by the CRAC systems from multiple regions, its MAT constraint can be relaxed in each region where it is associated with a CRAC system. The relaxation in each iteration is as follows. For a

prediction horizon k, the predicted inlet temperature $\hat{T}_{l,k}$ of the l-th ungrouped server can be computed based on the solution in the previous iteration. If the predicted temperature $\hat{T}_{l,k}$ exceeds the upper bound $\widetilde{T}_{l,k}$ given by Eq. (2), the deviation $\hat{T}_{l,k} - \widetilde{T}_{l,k}$ characterizes the amount of extra heat that needs to be removed from the l-th server inlet [8]. PTEC allocates this extra heat to each region g proportionally based on the total TCI of all CRAC systems in g. This is achieved by relaxing the MAT constraint for the l-th ungrouped server in each region g as $\widetilde{T}^g_{l,k} = \hat{T}_{l,k} - \frac{\sum_{j\in C_g}\gamma_{l,j}}{\sum_{j\in C}\gamma_{l,j}} \cdot (\hat{T}_{l,k} - \widetilde{T}_{l,k})$. Problem 1 is solved for each region g with the relaxed MAT constraint $\widetilde{T}^g_{l,k}$ for each ungrouped server l. The above process is repeated until all ungrouped servers' thermal requirements are satisfied. The pseudocode of the online stage and its convergence proof are omitted due to space constraints and can be found in [11].

In the partition-based predictive controller, Problem 1 is solved for each small region, resulting in significantly lower computation overhead compared with that of solving Problem 1 for the whole data center. In Section VII-C, we will evaluate the overhead and effectiveness of this approach.

VI. Implementation

A. Testbed and Sensor Deployment

We implemented PTEC on a single-rack testbed shown in Fig. 9. It consists of a rack of 15 1U servers in a 5×6 square feet room insulated by foam boards. On the rack, 15 servers are grouped every three servers with a 2U distance between every adjacent two groups. Each server is equipped with 2 PWM-controlled fans (Delta Electronics BFB1012EH) to cool the internal components. Each fan consumes a maximum of 29.4 W input power and the two fans contribute up to 25% of total power consumption of a server. Each server also has three on-board temperature sensors to monitor the CPU and server inlet temperatures. A Tripp Lite portable AC (SRCOOL12K) with a rated power of 3.5 kW is placed aside the server rack within the room. To enable its automatic control, we connect it to several power relays, which can be remotely switched by a program. Its return hot air register faces the side of server outlets, drawing the hot air. It delivers cold air through a register located at the bottom of the room in front of the rack, which is consistent with the popular raised floor cooling design in production data centers. However, due to this AC's limited cooling capacity, up to 6 servers can be running at the same time in our experiments. A total of 15 Iris temperature sensors are mounted with brackets at the outlets of the 5 group of servers. To monitor the AC status, we place an Iris temperature sensor at the AC cold air register and another at the hot air return register. To measure the power consumption of the servers and AC, we attach a wireless power meter for each server and AC. This small testbed allows us to study the fine-grained performance of PTEC.

B. System Implementation

Our system consists of two data collection networks and a base station that runs the predictive controller. The base station collects the data from the wireless sensors connected via 802.15.4 wireless links and the internal server sensors (inlet temperature sensor and fan speed sensor) over the Ethernet.

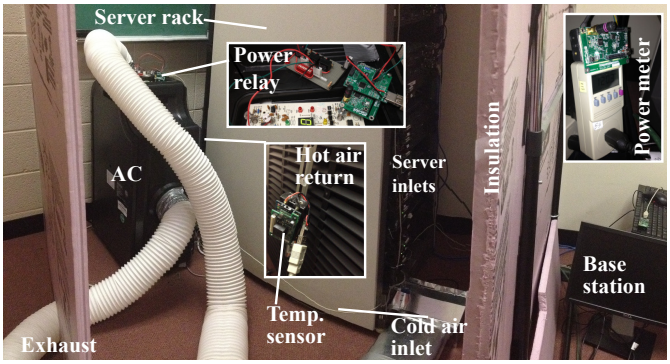

Fig. 9. A single-rack testbed that consists of a base station, a portable AC, a rack of 15 servers, and a total of 23 temperature/power sensors.

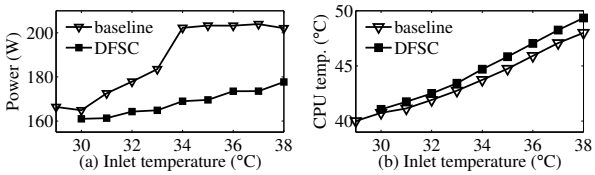

Fig. 10. Server power and CPU temperatures under DFSC and the baseline approach when the server is idle.

It also runs the optimization algorithm and sends the control commands to the AC. The data collection is implemented with JAVA on the base station, while the temperature prediction and the predictive controller are implemented with MATLAB.

Sensor network. We use a single-hop network architecture, where the base station sends the data collection requests to the sensors sequentially and each sensor transmits the measurements back. Every 30 seconds, the base station performs a round of sequential data collection from all sensors. Note that a multi-hop network topology can be used when more server racks are monitored. As this collection scheme works in a time-division fashion, the system experiences few collisions between the data transmissions of different sensors. The programs on all wireless sensors are implemented in TinyOS 2.1.

Server network. CPU utilization, on-board temperatures, fan speeds and DFSC settings are important thermal variables from each server. Data centers typically run various server monitoring tools (e.g., `atop`, `ganglia`) that can collect on-board sensor information. We implement a program to control and measure the CPU utilization, and report on-board temperatures and fan speeds from the `lm-sensors` utilities. The base station leverages the existing Ethernet infrastructure to collect these on-board sensor readings and DFSC settings.

Fan speed and AC control. A `GNU BASH` script running on each server implements the DFSC algorithm. A separate wireless connection is established between the base station and a TelosB mote that connects to a power relay control circuit board of the AC. When the mote receives the control signal from the base station, it turns on/off the AC.

VII. Performance Evaluation

We evaluate the performance of PTEC with testbed experiments in Section VII-A and VII-B2, and trace-driven Computational Fluid Dynamics simulations in Section VII-C.

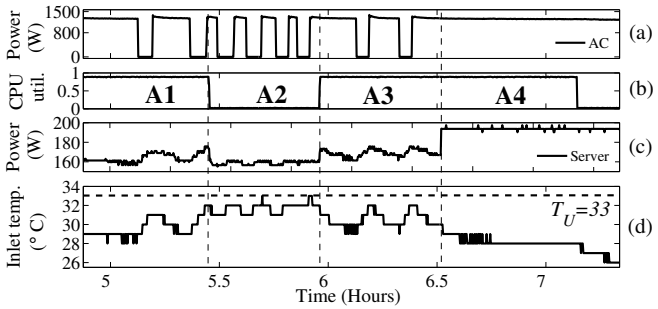

Fig. 11. Evolution of PTEC and *max cooling* baseline on a server. (a) AC power; (b) CPU utilization; (c) Server power excluding non-idle CPU power; (d) Server inlet temperature.

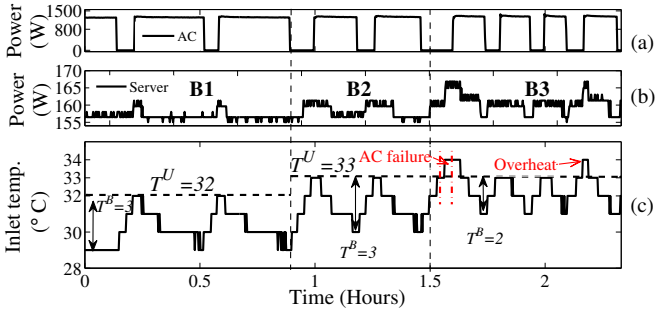

Fig. 12. Reactive control approach when the servers are idle. (a) AC power; (b) Server power excluding non-idle CPU power; (c) Server inlet temperature. Three periods are marked from B1 to B3.

TABLE I
AVERAGE POWER CONSUMPTION (WATT)

	A1	A2	A3	A4
AC power	903	639	931	1254
Server power	1065	1035	1077	1279

A. Effectiveness of DFSC

DFSC adopts two settings, i.e., $r_1 = \langle 28°\text{C}, 34°\text{C}\rangle$ and $r_2 = \langle 30°\text{C}, 42°\text{C}\rangle$. We employ a baseline approach that controls the fan speed solely based on the setting r_1 to meet the MAT requirement when the server is fully utilized. This baseline is consistent with the static fan speed control scheme used in most servers. As both algorithms adopt r_1 when the CPU is fully utilized, we compare them when the server is idle. The two curves in Fig. 10(a) show the server power consumption under the two approaches. The server power consumption of the baseline increases with inlet temperature much faster than DFSC. It reaches about 200 W when the temperature is 34°C. In comparison, DFSC consumes less than 180 W at the temperature of 34°C. Therefore, each individual idle server can save more than 20 W if T_U is 34°C. Moreover, as shown in Fig. 10(b), the CPU temperatures under our DFSC approach are slightly higher than those under the baseline approach. However, they are still significantly lower than the maximum allowed temperature of the CPUs (69°C). This result shows that DFSC can save significant amount of energy on the idle or low utilized servers.

B. Effectiveness of Predictive Controller

In this experiment, we compare PTEC with two baselines. The first baseline, referred to as *max cooling*, sets a fixed low CRAC setpoint while the server fans maintain the full speed. This baseline provides the maximum cooling capability to prevent overheating. The second baseline, referred to as *reactive control*, applies DFSC to control server fans and a hysteresis principle to control the AC reactively, so that the inlet temperatures are maintained within a range. Let $R(T^U, T^B)$ denote a reactive control strategy specified by two parameters, i.e., the temperature upper bound T^U and the temperature band T^B. Specifically, when the inlet temperature exceeds T^U, the AC starts to work until the inlet temperature is reduced to $T^U - T^B$. Then, the AC is turned off.

1) Comparison with max cooling: Fig. 11 shows the comparison between PTEC and the max cooling scheme. The optimization horizon $K = 12$ (6 minutes), the control cycle length $m = 2$ (1 minute), and the MAT constraint $T_U = 33°\text{C}$. We intentionally use a large RSD_U to study the effect of MAT constraint. The entire experiment comprises four periods, i.e., A1 to A4. Periods A1 to A3 run PTEC and Period A4 runs the max cooling. In Period A1, the CPU is fully utilized (Fig. 11(b)), which generates significant heat and may cause fast temperature rise. In response to this high CPU utilization, DFSC configures server fans to r_1, resulting in an average server power consumption of 170 W (Fig. 11(c)). Note that, in Fig. 11(c), CPU power consumption is not included and the power fluctuations are mostly caused by the fan speed changes. In Fig. 11(d), we can see that the MAT constraint is satisfied. When the system enters Period A2, the server switches to idle state. With lower CPU utilization, DFSC configures server fans to r_2, resulting in a relatively low server power consumption (Fig. 11(c)). Since the CPU utilization is low in Period A2, the system maintains a higher inlet temperature without overheating the servers. Thus, the AC can be turned off more frequently to save energy. In Period A3, the server CPU utilization becomes high again and the inlet temperature is maintained at a lower level. In Period A4, we apply the max cooling approach. Table I shows the average AC and server power consumption during each period. PTEC in Period A1 (i.e., full CPU utilization) and A2 (i.e., server idle) reduces total power consumption by 22% and 34%, respectively, compared with the max cooling in Period A4.

2) Comparison with reactive control: Fig. 12 shows the experiment results for the reactive control. We start the control with $T^U = 32°\text{C}$ and $T^B = 3°\text{C}$. At about 50 minutes, we increase T^U to 33°C. Under both settings, the inlet temperature remains below T_U. After 1.5 hours, we set $T^B = 2°\text{C}$. An unexpected AC failure occurred in this experiment, during which the AC failed to respond to the turning-on request due to a wireless link disconnection. As a result, the servers increase the fan speeds to respond to the increasing inlet temperatures caused by the failure. In Fig. 13(c), the higher power consumption indicates the higher fan speed. However, this failure lasts for 3 minutes only and does not affect the rest of the experiment. After about 2.3 hours, the temperature exceeds T_U for 2 minutes even if the AC has been turned on to react to overshooting T^U. Thus, for the reactive control approach to cope with the dynamic heat generated in a data center, T^U should be sufficiently low and T^B should be sufficiently large. Such conservative settings often lead to overcooling and excessive power consumption.

Fig. 13 shows the results of the inlet temperature of a server under PTEC and the reactive control approach with $T_U = 33°C$.

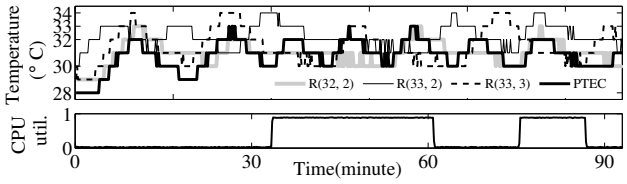

Fig. 13. Inlet temperature under PTEC and reactive control. $R(T^U, T^B)$ denotes a reactive control baseline with settings T^U and T^B.

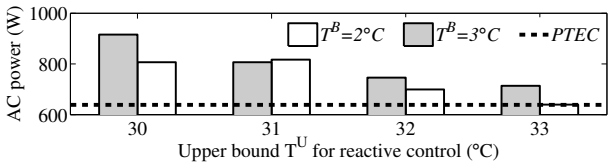

Fig. 14. AC power consumption of reactive control baselines and PTEC when the server is idle.

Control cycle length is set to $m = 6$ (3 minutes). It can be seen that PTEC consistently maintains the temperature below T_U. The reactive control with $T^U = 33°$, however, exceeds the MAT constraint T_U frequently. By lowering the T^U to 32°C, the reactive control can maintain the temperature below T_U. Note that as all approaches adopt the DFSC, their fan power consumptions are similar. Fig. 14 shows the AC power consumption of PTEC and the reactive control approach when the servers are idle. PTEC can reduce the power by up to 30%. In addition, PTEC also reduces the power by up to 20% when the servers are fully utilized.

C. Trace-Driven Computational Fluid Dynamics Simulations

In addition to the testbed evaluation, we study the performance of PTEC by Computational Fluid Dynamics (CFD) simulations driven by real workload data traces collected from the High-Performance Computing Center (HPCC) of Michigan State University. We use a commercial CFD software (ANSYS Fluent) to model a part of this data center, which hosts five racks of totally 229 servers. The racks are arranged in two rows with a cold aisle between them. Two in-row CRAC systems are installed for each row. Due to the lack of a complete CFD model of HPCC, we model each CRAC system as a cooling unit with three discrete cooling powers, i.e., 24 kW (rated power of each CRAC system in HPCC), 17 kW, and 9.6 kW. Thus, PTEC selects a cooling power instead of tuning temperature setpoint. Moreover, each CRAC system has two blower speeds. The simulations are driven by the server workload traces of these 229 servers. A time step is 5 minutes. Other settings are: $m = 4$ (20 minutes), $K = 8$ (40 minutes), and $w = 18$ (1.5 hours).

Fig. 15(a) shows the results of server inlet temperatures under PTEC. Initially, we set $T_U = 26$°C. We can see that PTEC controls the server inlet temperatures at around 26°C. At about 1.3 hours and 2.3 hours, we set $T_U = 22$°C and $T_U = 26$°C, respectively. All inlet temperatures are maintained at around T_U. Fig. 15(b) shows the resulting CRAC power consumption. After we increase T_U from 22°C to 26°C at about 2.3 hours, PTEC controls the CRAC systems to gradually increase the server inlet temperatures without violating the RSD requirement, and the CRAC power consumption is reduced by up to 35%.

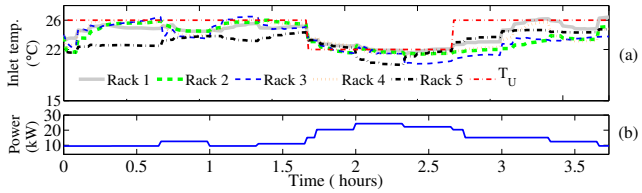

Fig. 15. PTEC in CFD simulations. (a) Average server inlet temperatures of each server rack; (b) CRAC power.

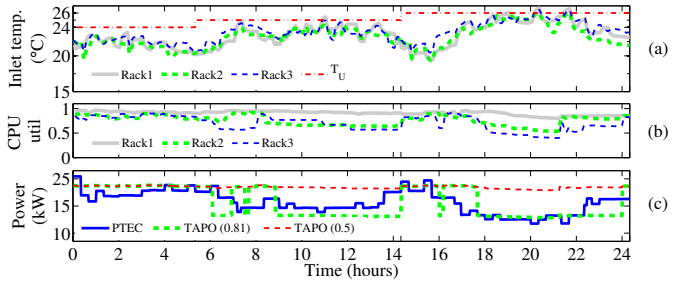

Fig. 16. Temperature control under dynamic CPU utilizations.

We now evaluate the effectiveness of PTEC under dynamic CPU utilization. The real CPU utilization traces span 96 hours. We select a portion of 24 hours that exhibit the highest dynamic levels to drive the simulation. For clear illustration, Fig. 16 shows the results of only three racks. The average CPU utilizations of these three racks are shown in Fig. 16(b). Initially, we set $T_U = 24°C$. Since the average utilizations of all three racks are high, PTEC keeps relatively low inlet temperatures to prevent overheating. At the 5.2 hours, we increase T_U to 25°C. Since the CPU utilization is still high, PTEC cannot further increase the inlet temperatures. After about 6 hours, the CPU utilization of Rack3 drops significantly. PTEC is then able to increase the inlet temperature without violating the new T_U. After 8 hours, PTEC reduces the inlet temperatures in response to the increased utilization of Rack3. After about 14 hours, we set $T_U = 26°C$. Initially, the inlet temperatures are maintained at a low level due to the high CPU utilization. After 16 hours, PTEC gradually increases the inlet temperatures close to T_U in response to the reduced CPU utilizations. This experiment shows that PTEC can well adapt to the dynamics of realistic data center server workload.

We also compare PTEC with a baseline approach that is a variant of an existing representative control approach TAPO [14]. TAPO uses a fixed low CRAC temperature setpoint T_L if the CPU utilization is higher than a predefined threshold u. Otherwise, it uses a fixed high CRAC temperature setpoint T_H. In our simulations, we set $T_L = 22$°C and $T_H = 26$°C. In [14], the threshold u is 0.5. Under this setting, as shown in Fig. 16(c), TAPO always uses T_L since the CPU utilizations in the simulation never drop below 0.5. By setting $u = 0.81$, the cooling power consumption under TAPO is comparable to that under PTEC. This result shows that the setpoints of TAPO need to be manually tuned to achieve the desirable performance. As the CPU utilization is unpredictable in real data centers, TAPO may not well adapt to dynamic CPU utilization.

We finally evaluate the partition-based algorithm in Section V-E. We partition HPCC to four regions, each of which contains one CRAC system. We compare our approach with a *brute-*

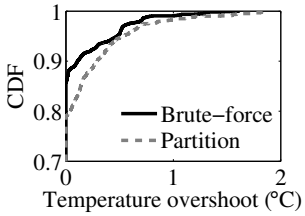

Fig. 17. CDF of temperature overshoot.

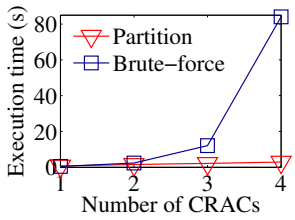

Fig. 18. Average execution time vs. the number of CRACs.

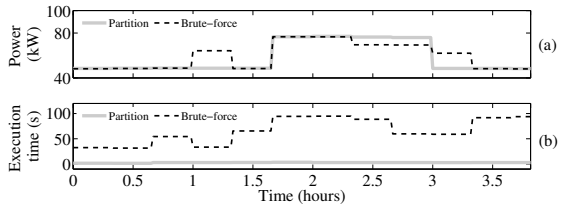

Fig. 19. Performance comparison between the brute-force and partition-based approaches. (a) power; (b) execution time.

force approach that exhaustively searches the optimal solution to Problem 1 for the whole data center. Fig. 17 shows the Cumulative Distribution Function (CDF) of the inlet temperature overshoots over T_U in a 12-hour simulation. For the brute-force approach, 90% of temperatures do not exceed T_U, and more than 95% of all temperatures fall within 1°C above T_U. The performance of our approach is slightly lower than the brute-force approach. As shown in Fig. 17, for a temperature overshoot of 1°C, the two approaches are comparable. In practice, we can set 1°C safety margin between the setpoint and the overheating temperature. Fig. 19 shows that our approach achieves comparable total power consumption and reduces the execution time significantly, compared with the brute-force approach. The average execution time of our approach is only 5% of that of the brute-force approach. Moreover, Fig. 18 shows that the execution time on an Intel Core i7-2600K 3.4 GHz CPU of the brute-force approach increases exponentially with the number of CRAC systems. On the contrary, the execution time of our approach increases slowly and linearly with the number of CRACs. These results show that our approach can find near-optimal solutions with satisfactory scalability. The low computational overhead enables PTEC to be implemented on portable hardware without relying on the computing infrastructure of monitored data center.

VIII. Conclusion

This paper presents the design and evaluation of PTEC – a system for predictive thermal and energy control in data centers. PTEC leverages the server built-in sensors and monitoring utilities, as well as a wireless sensor network to monitor the thermal and power conditions of a data center. Based on the sensor data, it predicts the server temperatures in real time, and optimizes temperature setpoints and cold air supply rates of cooling systems, as well as the speeds of server internal fans, to minimize their overall energy consumption. Moreover, PTEC enforces a set of thermal safety requirements including the upper bounds on server inlet temperatures and their variations, to prevent server overheating and reduce server hardware failure rate. Experiments on a small hardware testbed and trace-driven CFD simulations based on a production data center show that PTEC can reduce the cooling and circulation energy consumption by up to 34% and 30%, compared with an overcooling strategy and a reactive control strategy, respectively.

Acknowledgment

This research was supported in part by the U.S. National Science Foundation under grants CNS-0954039 (CAREER Award), CNS-1218475, NS-1218154 and CNS-1143607 (CAREER Award), in part by Singapore's Agency for Science, Technology and Research under the Human Sixth Sense Programme.

References

[1] http://www.google.com/about/datacenters/.
[2] ASHRAE 2011 thermal guidelines for data processing environments.
[3] Fancontrol. http://linux.die.net/man/8/fancontrol.
[4] Uptime institute 2012 data center industry survey, 2012.
[5] Uptime institute 2013 data center industry survey, 2013.
[6] A. Banerjee, T. Mukherjee, G. Varsamopoulos, and S. K. S. Gupta. Cooling-aware and thermal-aware workload placement for green hpc data centers. In *IGCC*, 2010.
[7] C. E. Bash, C. D. Patel, and R. K. Sharma. Dynamic thermal management of air cooled data centers. In *Thermal and Thermomechanical Phenomena in Electronics System*, 2006.
[8] P. Böckh and T. Wetzel. *Heat transfer: basics and practice*. 2012.
[9] W. Brendel and S. Todorovic. Segmentation as maximum-weight independent set. In *Conf. Neural Information Processing Systems*, 2010.
[10] J. Chen, R. Tan, Y. Wang, G. Xing, X. Wang, X. Wang, B. Punch, and D. Colbry. A high-fidelity temperature distribution forecasting system for data centers. In *RTSS*, 2012.
[11] J. Chen, R. Tan, G. Xing, and X. Wang. PTEC: A system for predictive thermal and energy control in data centers. Technical Report MSU-CSE-14-9, Computer Science and Engineering, Michigan State University.
[12] N. El-Sayed, I. Stefanovici, G. Amvrosiadis, A. A. Hwang, and B. Schroeder. Temperature management in data centers:why some (might) like it hot. In *Sigmetrics*, 2012.
[13] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper. Workload analysis and demand prediction of enterprise data center applications. In *Intl. Symp. Workload Characterization*, 2007.
[14] W. Huang, M. Allen-Ware, J. B. Carter, E. Elnozahy, H. Hamann, T. Keller, C. Lefurgy, J. Li, K. Rajamani, and J. Rubio. TAPO: Thermal-aware power optimization techniques for servers and data centers. In *IGCC*, 2011.
[15] M. Jonas, R. R. Gilbert, J. Ferguson, G. Varsamopoulos, , and S. Gupta. A transient model for data center thermal prediction. In *IGCC*, 2012.
[16] E. K. Lee, H. Viswanathan, and D. Pompili. VMAP: Proactive thermal-aware virtual machine allocation in hpc cloud datacenters. In *HiPC*, 2012.
[17] S. Li, T. Abdelzaher, and M. Yuan. TAPA: temperature aware power allocation in data center with map-reduce. In *IGCC*, 2011.
[18] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser. Renewable and cooling aware workload management for sustainable data centers. In *Sigmetrics*, 2012.
[19] J. Moore, J. Chasey, P. Ranganathanz, and R. Sharmaz. Making scheduling"cool": Temperature-aware workload placement in data centers. In *USENIX Annual Tech. Conf.*, 2005.
[20] L. Parolini, N. Tolia, B. Sinopoli, and B. Krogh. A cyber-physical systems approach to energy management in data centers. In *ICCPS*, 2010.
[21] L. Ramos and R. Bianchini. C-oracle: Predictive thermal management for data centers. In *HPCA*, 2008.
[22] N. Tolia, Z. Wang, P. Ranganathan, C. Bash, M. Marwah, and X. Zhu. Unified thermal and power management in server enclosures. In *InterPACK*, 2009.
[23] U.S. Environmental Protection Agency. Report to congress on server and data center energy efficiency, 2007.
[24] B. W. Wah, Y. Chen, and T. Wang. Simulated annealing with asymptotic convergence for nonlinear constrained optimization. *J. Global Optimization*, 39(1):1–37, 2007.
[25] Z. Wang, C. Bash, N. Tolia, M. Marwah, X. Zhu, and P. Ranganathan. Optimal fan speed control for thermal management of servers. In *IPACK*, 2009.
[26] M. Zapater, J. L. Ayala, J. M. Moya, K. Vaidyanathan, K. Gross, and A. K. Coskun. Leakage and temperature aware server control for improving energy efficiency in data centers. In *Conf. Design, Automation and Test in Europe*, 2013.
[27] R. Zhou, C. Bash, Z. Wang, A. McReynolds, T. Christian, and T. Cader. Data center cooling efficiency improvement through localized and optimized cooling resources delivery. In *Intl. Mechanical Engineering Congress and Exposition*, 2012.