

# Towards Optimal Sensor Placement for Hot Server Detection in Data Centers

Xiaodong Wang<sup>†</sup>, Xiaorui Wang<sup>†</sup>, Guoliang Xing<sup>‡</sup>, Jinzhu Chen<sup>‡</sup>, Cheng-Xian Lin\* and Yixin Chen<sup>¶</sup>

<sup>†</sup>Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, USA

<sup>‡</sup>Department of Computer Science and Engineering, Michigan State University, USA

\*Department of Mechanical, Aerospace and Biomedical Engineering, University of Tennessee, Knoxville, USA

<sup>¶</sup>Department of Computer Science and Engineering, Washington University in St. Louis, USA

<sup>†\*</sup>{xwang33, xwang, lincx}@utk.edu   <sup>‡</sup>{glxing, chenjinz}@cse.msu.edu   <sup>¶</sup>chen@cse.wustl.edu

**Abstract**—Recent studies have shown that a significant portion of the total energy consumption of many data centers is caused by the inefficient operation of their cooling systems. Without effective thermal monitoring with accurate location information, the cooling systems often use unnecessarily low temperature set points to overcool the entire room, resulting in excessive energy consumption. Sensor network technology has recently been adopted for data-center thermal monitoring because of its non-intrusive nature for the already complex data center facilities and robustness to instantaneous CPU or disk activities. However, existing solutions place sensors in a simplistic way without considering the thermal dynamics in data centers, resulting in unnecessarily degraded hot server detection probability. In this paper, we first formulate the problem of sensor placement for hot server detection in a data center as a constrained optimization problem. We then propose a novel placement scheme based on Computational Fluid Dynamics (CFD) to take various factors, such as cooling systems and server layout, as inputs to analyze the thermal conditions of the data center. Based on the CFD analysis in various server overheating scenarios, we apply data fusion and advanced optimization techniques to find a near-optimal sensor placement solution, such that the probability of detecting hot servers is significantly improved. Our empirical results in a real server room demonstrate the detection performance of our placement solution. Extensive simulation results also show that the proposed solution outperforms a commonly used placement solution in terms of detection probability.

## I. INTRODUCTION

Power and thermal management has become a key challenge in the design of large-scale data centers. In a 2007 report to the US Congress [1], the Environmental Protection Agency (EPA) estimated that the annual data center energy consumption in the US will grow to over 100 billion kWh at a cost of \$7.4 billion by 2011. One of the key reasons for data centers to have excessive energy consumption is the inefficient operation of their cooling systems (*e.g.*, a set of Computer Room Air Conditionings (CRAC)), which can account for up to half of their energy consumption [1]. Because of the lack of visibility in the operating conditions of data centers, the cooling systems often use unnecessarily low temperature set points to reduce the danger of creating any hot spot. However, without effective thermal monitoring with accurate location

information, excessive energy consumption is often caused by overcooling the entire space [2]. In addition to high electricity bills and negative environmental implications, the high heat dissipation, caused by server overheating, may also lead to the malfunction of many hardware components and even the shutdown of entire data centers. For example, the most used online encyclopedia, Wikipedia, went down on July 5th, 2010, because the failure of a cooling unit caused server overheating and then the power outage of the data center [3].

To prevent thermal emergencies in a data center with reduced cooling energy costs, a variety of thermal control schemes have been recently proposed. For example, HP researchers developed an Adaptive Vent Tile (AVT) technology [4] to automatically adjust mechanical louvers mounted to the floor vent tiles so that detected hot areas can get more cool air. Bash et al. [5] proposed to adaptively adjust the temperature set point and air flow rate (*i.e.*, fan speed) of each individual CRAC in a data center. Other control algorithms (*e.g.*, [6], [7]) have also been proposed to handle detected hot servers by throttling their CPU frequencies, migrating away their workloads, or shutting them down if necessary. However, in order to be effective, all the existing control solutions must work with certain thermal monitoring mechanisms to ensure that the locations of hot areas and servers can be accurately detected and the information can be sent to the controllers promptly. For instance, in the Wikipedia example mentioned above, if the hot areas created by the failed cooling unit had been more accurately detected, the blackout could have been avoided by effectively directing the air flows of remaining CRAC units to cool down the overheating servers.

Efficient thermal monitoring is challenging, given the data centers' complex air flow and thermal dynamics. Traditionally, simple thermostats or wired temperature probes are used to provide coarse-grained thermal monitoring, which cannot effectively support the monitoring granularity required by the thermal control schemes to conduct energy-efficient cooling. Wireless Sensor Network (WSN) technology has recently been identified as an ideal candidate for data-center thermal monitoring [8][5] due to several of its salient advantages. First, it can provide good coverage with accurate localization for *global* thermal management decisions in a data center. Second, it is non-intrusive, as the sensors use wireless communications

This work was supported, in part, by NSF under CAREER Award CNS-0845390 and Grants CNS-0720663, CNS-0915959, CCF-1017336, and CNS-0954039, and by Microsoft Research under a Power-Aware Computing Award.

and thus require no additional network and facility infrastructure in an already complicated data-center environment. Third, compared to the thermal sensors on motherboards, the wireless sensors are less sensitive to instantaneous CPU or disk activities, leading to less noisy thermal readings [8]. It is also important to note that the wireless sensors used for server thermal monitoring do not rely on batteries, because plenty of power resource is provided in data centers.

Although WSN technology has shown promise in data-center thermal monitoring, an important issue that has been overlooked by existing solutions is how to optimally place sensors in a data center such that all the possible overheating locations are well covered and monitored with maximized detection probabilities. Currently, many real data centers just simply place the same number of sensors on each rack uniformly at a constant distance from each other, without considering the thermal dynamics in the data center. For example, in a real data center located in HP Labs in Palo Alto, five sensors are placed on the front side of each rack from the top to the bottom to keep the inlet temperature at or below 24°C for all running servers [6]. Five sensors are used for each rack because it is usually preferable not to put too many wireless sensors on a rack for the considerations of space and cost, due to the very dense installation of high-density servers (e.g., up to 128 blade servers per rack). In addition, a highly dense deployment of sensors may cause the wireless network to have significantly increased levels of channel contention and thus, unacceptably long communication delays [9][10]. However, such a simplistic sensor deployment strategy may result in an unnecessarily degraded detection probability. In contrast, an optimized placement solution can intelligently place sensors based on the systematic analysis of the thermal dynamics in the data center, by considering the locations of the CRAC systems and the server racks, as well as the rack layout and various air flows in the room. As a result, better coverage can be achieved for servers that have a greater potential to become overheating. Consequently, given the same number of sensors, such an optimized solution can lead to a significantly improved detection probability and so a better chance for the existing control schemes to prevent thermal emergencies.

In this paper, we propose a novel sensor placement scheme for improved hot server detection performance, which can enhance the thermal control operations in data centers. Our placement scheme is developed based on the analytical results from Computational Fluid Dynamics (CFD), a powerful mechanical fluid dynamic analysis approach. CFD is widely used to analyze the fluid dynamics in various engineering fields, such as aircraft engine design and environmental analysis for buildings. CFD has already been used by data center designers to make intelligent decisions on layout design and rack deployments, but *not yet* for sensor deployments. In this paper, we use CFD to model the thermal environment of a given data center under different thermal emergency conditions and apply interpolation techniques to improve the thermal analysis results from CFD. Accordingly, for a given number of sensors, we seek to place them in the data center so that

the potential overheating servers (due to workload increases, CRAC failures, etc) at any location can be detected with the maximum detection probability. We formulate this problem as a constrained optimization problem based on data fusion techniques to allow sensors to make collaborative detection decisions of server overheating. Based on the formulation and the CFD analysis, we design a heuristic algorithm to find a near-optimal placement solution with a significantly reduced computational complexity, despite a huge search space.

Specifically, the contributions of this paper are four-fold.

- While the current WSN-based thermal monitoring solutions in many real data centers rely on simplistic sensor deployment without considering the thermal dynamics in the data center, we propose a novel sensor placement scheme to intelligently place sensors for maximized hot server detection probabilities.
- We propose to use CFD to model the thermal dynamics of a data center in various overheating scenarios (e.g., different servers are overheating due to workload increases or CRAC failures). CFD analysis provides a theoretical foundation for our sensor placement solution. We apply interpolation techniques to further refine the analytical results from CFD.
- We formulate optimal sensor placement as a constrained optimization problem and propose a heuristic algorithm to find a near-optimal solution with a significantly reduced computational complexity.
- We evaluate our sensor placement scheme in a real server room with 13 racks and more than 100 servers. Both our empirical and simulation results demonstrate that our placement solution can significantly improve hot server detection performance.

The remainder of this paper is organized as follows. Section II highlights the distinction of our work by discussing related work. Section III presents the data fusion model we used and the formulation of the hot server detection problem in data centers. Section IV introduces the fundamentals of the Computational Fluid Dynamics approach and provides an example of how to model a server room in CFD. Section V elaborates on how to use the analytical results from CFD in our sensor placement problem and proposes a heuristic algorithm to solve the problem. In Section VI, we evaluate our sensor placement scheme using simulations and in a real server room. Section VII concludes the paper and discusses the possible future work.

## II. RELATED WORK

Thermal management in data centers has been widely studied in the past. Moore et al. have proposed a temperature-aware workload placement scheme for data center [7]. Optimization schemes for data center thermal management using model-based approaches have been proposed in [11][12]. An automated, online, predictive thermal management scheme for data centers is also proposed in [13]. However, none of the above mentioned studies has explored the possibility of using wireless sensor networks. Several projects have adopted

sensor networks in data center for temperature monitoring. For example, a hybrid wired and wireless sensor network is used in [8] for data center thermal monitoring. A sensor network is also used in [5] to manipulate conventional CRAC units within the data center. However, none of the past methods addresses how to intelligently deploy sensors to improve the hot server detection performance. Our work is different from all the aforementioned research. We not only explore the benefits of using wireless sensor networks for temperature monitoring in data centers, but also maximize the detection probability of the potential overheating servers. It is important to note that our sensor placement solution is complementary to existing thermal control scheme (such as [4][5][6][7]) because more accurate thermal monitoring can significantly enhance the performance of thermal control. For example, as shown in [5], even a very simplistic deployment of temperature sensors can lead to a 50% saving in cooling energy.

Computational Fluid Dynamics (CFD) has been used to model the data center operating environment and server rack operating conditions. Patel et al. have used CFD in [14] to model and analyze the air temperature specification in the data center. Impact of CRAC failures on static provisioning has also been studied using CFD models in [15]. Jeohwang et al. have modeled the thermal profile for an operating rack in detail in [16] to provide a bridge between the individual component thermal status and data center thermal profile. Different from all the previously mentioned studies, our paper uses CFD to model different thermal emergency situations in data centers when servers (or racks) are overheating at any possible locations. We then use the CFD modeling results to guide the sensor deployment for the various overheating conditions, such that any thermal emergency associated with server workload dynamics or CRAC failures can be effectively monitored and reported by the sensor network.

Target detection and monitoring is one of the most important tasks of wireless sensor networks. Several existing projects have explored how to deploy sensors effectively to improve the detection and monitoring performance. A sensor placement scheme based on the Multivariate Gaussian Process model is proposed in [17], which provides most informative results after the data training period. A fast sensor placement approach for fusion-based target detection is proposed in [18] to minimize the number of deployed sensors while achieving assured detection performance. Different from these previous schemes of sensor deployment, the sensor deployment approach we propose leverages on the computational results from CFD which analyzes the thermal condition of a monitored field based on theoretical thermal dynamics. Furthermore, the model training approach proposed in [17] is not applicable for data center thermal emergency monitoring, because the thermal emergency scenario should not be created simply for the collection of training data.

### III. HOT SERVER DETECTION PROBLEM

In this section, we first introduce the hot server (hot spot) detection model in sensor networks. We then formally formu-

late the data center hot server detection problem.

#### A. Hot Server Detection Model

It is always desirable to cover as many locations as possible in the thermal monitoring of a data center. For example, we may want to monitor the temperature at the inlet or outlet of each rack or even each server. Therefore, the number of monitored locations in a large data center is usually large. It is therefore unrealistic to deploy an individual sensor for every monitored location, especially in a wireless sensor network environment. It is not only because of the space limitation and cost, but also because of the resulting high network density. A very dense wireless sensor deployment often leads to undesired network communication quality because of severe interferences and channel contentions among wireless transmissions [9][8]. To conduct data center temperature monitoring and hot server detection with a reasonable number of sensors, sensor nodes should collaborate with each other when making detection decisions. Data fusion [19][20][21], a widely adopted technique for improving the detection performance of sensor systems by collaboration, is well-suited in this scenario.

It is clear that temperatures at the locations far from a heat source are less likely to be correlated with this source. Therefore, we define a *fusion region* of each monitored location as a sphere with a *fusion radius*  $R$ , where the monitored location is the center of that sphere. The sensors within the fusion region of a monitored location should collaborate to make the detection decision for that location. We adopt a data fusion scheme which calculates the average temperature from all the sensors within the fusion region of the monitored spot and compare the average value with a detection threshold  $\eta$ . If the average temperature is larger than the threshold, the decision of a hot server detection is positive.

The measurements of a sensor are usually corrupted by noise. Denote the measurement noise strength measured by sensor  $i$  as  $N_i$ , which follows the zero-mean normal distribution with a variance of  $\sigma^2$ , i.e.,  $N_i \sim \mathcal{N}(0, \sigma^2)$ . The final measured temperature,  $T_m$ , from a sensor at location  $(x_i, y_i, z_i)$  can be presented as

$$T_m(x_i, y_i, z_i) = T_r(x_i, y_i, z_i) + N_i^2 \quad (1)$$

where  $T_r$  is the real temperature at that location without noise.

Assuming there are  $n$  sensors within the data fusion region of a monitored spot, the detection probability of the hot server existence at the monitored point can be calculated as

$$\begin{aligned} P_D &= P\left(\frac{1}{n} \sum_{i=1}^n (T_r(x_i, y_i, z_i) + N_i^2) > \eta\right) \\ &= 1 - P\left(\sum_{i=1}^n \left(\frac{N_i}{\sigma}\right)^2 \leq \frac{n\eta - \sum_{i=1}^n T_r(x_i, y_i, z_i)}{\sigma^2}\right) \end{aligned} \quad (2)$$

where  $\eta$  is the detection threshold of overheating. Because of the measurement noise from the sensor device,  $\eta$  is different from the real temperature threshold for a hot-spot, denoted as  $C$ . With a high noise level from the measurement, a detection

system is likely to give a false alarm when there is no real event. In our case, we define the false alarm rate when there is actually no hot server as follows

$$P_F = P\left(\frac{1}{n} \sum_{i=1}^n (N_i^2 + C) > \eta\right) \quad (3)$$

$$= 1 - P\left(\sum_{i=1}^n \left(\frac{N_i}{\sigma}\right)^2 \leq \frac{n(\eta - C)}{\sigma^2}\right)$$

We assume Gaussian Noise, *i.e.*,  $N_i/\sigma \sim \mathcal{N}(0, 1)$ . Therefore,  $\sum_{i=1}^n (N_i/\sigma)^2$  follows the Chi-square distribution [22] with  $n$  degrees of freedom, denoted as  $\chi_n(\cdot)$ . Hence, Equations (2) and (3) can be modified as follows

$$P_D = 1 - \chi_n\left(\frac{n\eta - \sum_{i=1}^n T_r(x_i, y_i, z_i)}{\sigma^2}\right) \quad (4)$$

$$P_F = 1 - \chi_n\left(\frac{n(\eta - C)}{\sigma^2}\right) \quad (5)$$

### B. Problem Formulation

We assume that there are  $M$  locations (e.g., blade servers) in the data center room for which we need to monitor temperature. Given a limited number of sensors,  $N < M$ , we need to find the placement of these  $N$  sensors such that we can detect the overheating emergency at any of the  $M$  locations with the highest possible confidence. Before presenting our formal formulation, we first introduce the following notation:

- $l_i$ , monitored location  $i$  with location coordinates  $(x_{li}, y_{li}, z_{li})$ .
- $P_{F_i}$ , the false alarm rate of reporting an overheating emergency at  $l_i$ .
- $P_{D_i}$ , the detection probability of an overheating emergency at  $l_i$ .
- $n_i$ , the sensor number within the fusion region of  $l_i$ .

Our goal is to maximize the average detection probability of all the monitored locations

$$\max \frac{1}{M} \sum_{1 \leq i \leq M} P_{D_i} \quad (6)$$

subject to the following constraint

$$P_{F_i} \leq \alpha \quad \forall 1 \leq i \leq M \quad (7)$$

where  $\alpha$  is the detection false alarm rate requirement. We note that the false alarm rate needs to be bounded in many practical scenarios due to the waste of system resources. Note that network connectivity can be another constraint to be considered in our formulation to ensure that every sensor is connected to the network for effective data fusion. However, due to the high-density installation of servers (e.g., blade servers) in the data center, our experience shows that connectivity is not a major concern, because the communication range of a wireless sensor is 30m indoors [23].

For a certain sensor placement,  $P_{F_i} \leq \alpha$  is a necessary condition in our problem. By Equation (5), we convert the

constraint in Equation (7) to  $\eta_i \geq \frac{\sigma^2 \chi_n^{-1}(1-\alpha)}{n_i} + C$ , a constraint for the detection threshold  $\eta$  at monitored location  $i$ , where  $\chi_n^{-1}(\cdot)$  is the inverse function of  $\chi(\cdot)$ . Using this equation, we can obtain the desired detection threshold based on the required false alarm rate and use it to calculate the detection probability. From Equation (4) we know that  $P_{D_i}$  decreases when  $\eta_i$  increases. Therefore, to maximize the detection probability, we remove the inequality in the constraint and only use the lower bound  $\alpha$ . Hence,  $\eta_i$  can be calculated as

$$\eta_i = \frac{\sigma^2 \chi_n^{-1}(1-\alpha)}{n_i} + C \quad (8)$$

## IV. CFD MODELING FOR DATA CENTER

In this section, we first introduce Computational Fluid Dynamics (CFD), the tool we use to analyze the thermal conditions in a data center. We then provide an example of how to model a server room in practice using Fluent [24], a widely used CFD modeling software package.

### A. CFD Modeling

CFD is a fluid mechanics approach that analyzes problems of fluid flows based on numerical methods and algorithms. The key for CFD modeling is to solve the governing transport equations represented in the following conservation law form:

$$\frac{\partial \rho \phi}{\partial t} + \frac{\partial \rho U_j \phi}{\partial x_j} = \frac{\partial}{\partial x_j} (\Gamma_{\phi,eff} \frac{\partial \phi}{\partial x_j}) + S_\phi \quad (9)$$

where  $\phi$  represents different parameters such as mass, velocity, temperature or turbulence properties;  $\rho$  is the fluid (air) density;  $t$  is the time for transient simulations;  $x_j$  is the coordinate variable for  $x$ ,  $y$  or  $z$  with  $j$  being 1, 2 or 3;  $U_j$  is the velocity in different directions;  $\Gamma$  is the diffusion coefficient; and  $S$  is the source for the particular variable. For example, when  $\phi$  is the air temperature,  $S$  stands for the volumetric heat rate from a source component. The four equation terms represent transient, convection, diffusion, and source parts of transport phenomenon taking place in the spatial domain [25].

The partial differential equations listed in Equation (9) represent a system, where all the transport equations are coupled together and require to be solved simultaneously. For a complicated environment, such as a data center, no closed-form solutions can be found for the airflow and heat transfer of the entire system. Therefore, the most fundamental consideration in CFD is how to treat a continuous fluid in a discretized fashion, such that numerical methods can be applied to find the solutions. Most CFD software packages apply the control volume method to find numerical solutions.

### B. Example of Server Room CFD Modeling

Using CFD to perform a continuous fluid model requires the discretization of the spatial domain. One method to achieve this goal is to discretize the spatial domain into small cells by generating volumetric grid. After the discretization, necessary boundary conditions and suitable algorithms need to be applied to solve the above-mentioned transport equations. Several popular software packages, such as Fluent, FloTHERM,

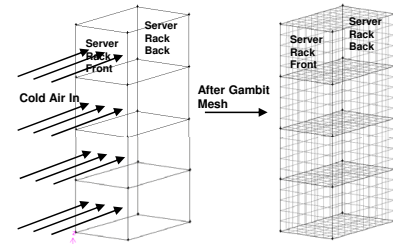
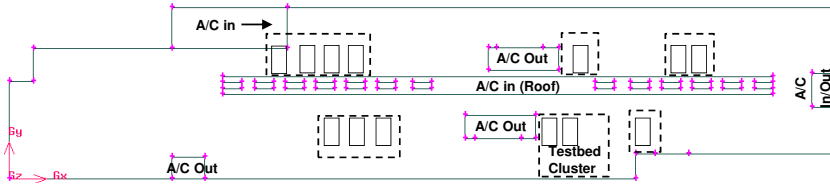


Fig. 1. Top view layout of the server room in the EECS department at the University of Tennessee. Fig. 2. Example of meshing a server rack in Gambit. The size is about  $30\text{m} \times 7\text{m} \times 3.4\text{m}$ . Small solid boxes represent server racks and the dotted boxes represent rack clusters.

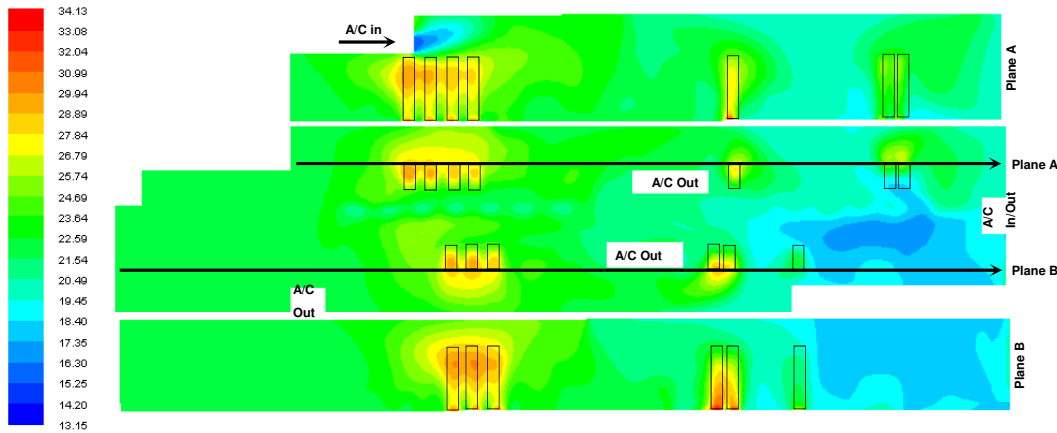


Fig. 3. Temperature ( $^{\circ}\text{C}$ ) map in the server room with all servers running at full power. The middle graph is the top view temperature map at plane of  $z = 0.5\text{m}$ . The top graph is the temperature map of cross section plane A. The bottom graph is the temperature map of cross section plane B. Solid-line boxes represent server racks.

FloVENT, and PHOENICS, can be used for CFD modeling purpose. In our project, we use Fluent, a widely used CFD software package from ANSYS Inc., to perform the spatial domain meshing and solution finding, respectively.

The CFD model we establish in this example is for the server room in the EECS department at the University of Tennessee, Knoxville, whose top view layout is shown in Figure 1. As introduced before, CFD has already been used by data center designers to make intelligent decisions on layout design and rack deployments, but not yet for sensor deployments. In this paper, our goal is to use CFD to model various server overheating scenarios, such that sensor placement can be guided for improved thermal emergency detection performance. In the first step, we use Gambit, which is a grid generator, to perform the geometry establishment for this room. Basically, we choose different geometric shapes and perform unification or split to establish the geometric model for the entire server room based on the real measured scales. Then we add different geometric shapes into the entire room geometry to model server racks and Computer Room Air Conditioning (CRAC) units according to their geographic location and corresponding scale. To simplify the complexity of the server room modeling, we group the servers in each rack into 4 blocks from top to bottom by equal physical size, as shown in Figure 2. This approach was also adopted by other projects, such as in [14], to simplify the geometric establishments in CFD modeling. We plan to handle more complex CFD modeling in our future work.

After all components are added into the geometric model, we need to specify different boundary types, such as walls, the inlets/outlets of air conditioning units, and interior faces within the entire room domain. The last step is to divide the entire geometric model into smaller scale cells by applying geometry meshing in Gambit. The grid size is a user-specific parameter. With a fine grid, more accurate CFD modeling can be reached. However, a fine grid increases the computational burden in later on stages, when solving the transport equations by numerical methods. Figure 2 shows an example of using Gambit to perform the geometric mesh for a server rack.

After meshing the entire server room geometrically in Gambit, we export the grid to the second software package, Fluent, to solve the transport equations in Equation (9). Fluent requires all the boundary conditions of our geometric model to be specified. For example, we need to specify the temperature of the incoming cold air from all the CRAC units. We also need to specify the power dissipation of each server rack (block), as well as the fluid velocity of each server block. The standard k-epsilon two-equation turbulence model is used to simulate the turbulent flow. The governing equations were solved with numerical schemes of secondary accuracy. Figure 3 shows a colored temperature map after solving the transport equations in Fluent. This is a scenario in which all the server racks are running with maximum power. We can see that the back sides of server racks are significantly hotter than the front sides.

## V. CFD-GUIDED SENSOR PLACEMENT

In this section, we introduce how to use the results from the CFD analysis to guide sensor placement, targeting the maximum hot server detection probability. We also design a heuristic algorithm for solving this detection probability maximization problem.

### A. Overview of Our Approach

Using CFD tools for our sensor placement primarily involves two steps. In the first step, we establish a geometric model for the data center room in Gambit, mesh the geometry and export the grid to Fluent. We then take measurements for the incoming cold air temperature and air flow rate from the inlet of every CRAC unit. These measurements, along with the power consumption of each block, are the input parameters to Fluent. In order to detect a potential overheating scenario, we set the power dissipation of each rack to the overheating scenario, one rack at a time, and solve the temperature distribution for each overheating scenario by an iterative solution procedure in Fluent.

In the second step, we feed the results from the CFD analysis of all the overheating scenarios to our optimization algorithm to find the best locations for sensor placement. For detection locations, we assume that our sensor placement needs to monitor the temperature of the center point at the back face of each block. To solve the placement problem efficiently, we develop a heuristic algorithm based on the Constrained Simulated Annealing approach [26]. Since the results from the CFD analysis are temperatures at discretized locations, our algorithm features a temperature interpolation approach [27] to interpolate the missing values from the CFD results. Both of these two approaches are explained in detail in the following sections.

### B. Spatial Temperature Interpolation

As explained in the previous section, the results from the CFD tools are temperatures at discretized locations. The granularity of the data set depends on the density of the grid in the previous geometric model establishment in Gambit. There is a trade-off between granularity and performance. A higher density grid means a finer grained granularity of the data and more accurate results. However, it also requires more significant computational resources, *i.e.*, computational time and memory size. Therefore, in our design, we choose a granularity of approximately 10cm for our model. This choice results in a one-time running expense of Fluent for less than two hours. For temperature-unknown locations, we choose to interpolate the data.

There are various kinds of spatial interpolation methods [28]. The measuring target in our project is temperature, which is spatially continuous and short range correlated. Therefore, we choose to use the Inverse Distance Weighting (IDW) [27] technique to interpolate the temperature. IDW method estimates the value of an attribute at unsampled points by linear combination of values at sampled points weighted by an inverse function of distance from the sampled point to

the interpolating target point. Denote the interpolation target point as  $l_0$  and the known sampled points as  $l_i$ , the detailed interpolation is calculated as

$$T(l_0) = \sum_{i=1}^n \lambda_i T(l_i) \quad (10)$$

where  $\lambda_i$  is the weight of each element, which can be expressed as

$$\lambda_i = \frac{1/d_i^p}{\sum_{i=1}^n 1/d_i^p} \quad (11)$$

where  $d_i$  is the distance between  $l_0$  and  $l_i$ ,  $p$  is a power parameter, and  $n$  is the number of sampled points used for the estimation. In our approach, we adopt the same range as the fusion radius  $R$  in Section III and include all the known temperature points from CFD within that range from the interpolating target point.

The assumption behind IDW is that sampled points closer to the unsampled interpolating target are more similar to the target than those further away points in their values. The main factor affecting the accuracy of IDW is the value of the power parameter. The choice of power parameter  $p$  is arbitrary. The most popular choice of  $p$  is 2 and the resulting method is often called the Inverse Square Distance (IDS). Hence, we adopt 2 as the power parameter.

### C. Lightweight Sensor Placement (LSP) Algorithm

Our goal is to find the optimal sensor placement locations to maximize the average detection probability of all the monitored locations. Since every sensor has three location coordinates, with  $N$  sensors to be placed in the domain, we need to solve a problem with  $3N$  variables. A straightforward approach to this problem is to solve this entire problem at once. This can be achieved by a nonlinear programming solver based on the Constrained Simulated Annealing (CSA) algorithm [26]. CSA is an extension of the conventional Simulated Annealing algorithm for solving the global constrained optimization problem with discrete variables. Theoretically, CSA can reach a global optimal solution by converging asymptotically to a constrained global optimum with a probability of 1. However, a limitation of CSA is that its computational complexity grows exponentially with respect to the number of variables and the solution search space [26][18]. The execution time of the algorithm can reach up to thousands of days with hundreds of sensors to place [18]. Therefore, it is not realistic to solve the entire placement problem as a whole. In this paper, we design a lightweight optimization algorithm based on CSA to reduce the complexity of the algorithm and find a near-optimal solution to our sensor placement problem.

First, we reduce the search space of the entire problem by dividing the entire search domain, *i.e.*, the server room, into several clusters. For example, from Figure 1 we see that servers can be grouped based on their geographical locations such that the distance between two groups is greater than the desired fusion range. It is not feasible to calculate a solution in CSA while some sensors are placed outside the fusion region of any monitored locations. Therefore, we group the

monitored locations into clusters according to the rule that for a monitored location  $l_i$  to be in a cluster, at least one other location in the same cluster should be within a  $2R$  distance of  $l_i$ .  $R$  is the fusion radius defined in Section III. The reason for choosing  $2R$  as the clustering parameter is that if a monitored location  $l_i$  is not within the  $2R$  range of any other monitored location in one cluster,  $C$ , no sensor can cover  $l_i$  and any location in  $C$  at the same time. Therefore,  $l_i$  should not be put in cluster  $C$ . After the clusters are formed, we calculate a new search space  $S_k$  for cluster  $C_k$  by the following equations

$$\begin{aligned} S_{ku_{min}} &= \min_{\forall l_i \in C_k} l_{iu} - R \\ S_{ku_{max}} &= \max_{\forall l_i \in C_k} l_{iu} + R \end{aligned} \quad (12)$$

where  $u$  is one of the coordinate subscripts,  $x, y$ , or  $z$ . Figure 1 shows an illustration of the clusters (dotted boxes) in which the search space is significantly reduced.

---

**Algorithm 1** LIGHTWEIGHT SENSOR PLACEMENT( $D$ )

---

**Input:** *Sensor number*  $N$ , *Cluster Information* :  $C_k, S_k, K, CFD_{data}$

**Output:** *Placement solution*  $D$

```

1:  $d_j = 0$  for all  $1 \leq j \leq K$ 
2:  $\bar{P}_j[N] = 0$  for all  $1 \leq j \leq K$ 
3: for  $i = 1$  to  $N$  do
4:    $\Delta P = 0$ 
5:   for  $j = 1$  to  $K$  do
6:      $n = d_j + 1$ 
7:     if  $n > n_j$  then
8:        $[\bar{P}_j[n], D_j[n]]$ 
9:        $= CSA(n, S_j, C_j, CFD_{data})$ 
10:       $n_j = n$ 
11:    end if
12:     $\Delta P' = n\bar{P}_j[n] - d_j\bar{P}_j[d_j]$ 
13:    if  $\Delta P' > \Delta P$  then
14:       $k = j$ 
15:       $\Delta P = \Delta P'$ 
16:    end if
17:  end for
18:   $d_k = d_k + 1$ 
19:   $D = \cup_{j=1}^K D_j[d_j]$ 
20: end for
21: return  $D$ 

```

---

We now introduce the heuristic algorithm for solving the placement problem, based on the clusters and the CSA solver. In our algorithm, we add sensors to the entire monitoring domain one by one. The question is which cluster should take in the new sensor. We calculate the increments of the global average detection probability by adding the new sensor to each particular cluster. The increments of average detection probability, when adding the new sensor to a specific cluster, is calculated by the CSA solver within the search space of that cluster. After calculating all the clusters with the new sensor added in, we choose the cluster that adds the most incremented global average probability and add the new sensor

to it. Note that in this step we do not need to calculate the result for every cluster, because the clusters that are not picked by the last sensor have already been calculated in the last round for the new sensor number. Therefore, the computational time is significantly reduced. Another reason that the LSP algorithm can decrease the computational complexity is that the benefit of adding a sensor to a specific cluster is greater when fewer sensors were previously in that cluster. Therefore, the algorithm favors adding sensors to clusters which have fewer sensors. This reduces the number of variables when using the CSA solver. The pseudo code of this algorithm is listed in Algorithm 1.

## VI. EXPERIMENTAL RESULTS

In this section, we first explain the setup of our experiments. We then evaluate our sensor placement approach with simulations and on a hardware testbed in a real server room.

We conduct our analysis and hardware experiment based on the CFD model of the server room shown in Figure 1. There are 13 racks in the server room. Each rack is divided into 4 blocks, with one monitored location at the center of the back face of each block, leading to 52 monitored locations in total. We use CFD to model the rack overheating scenario for all 13 racks one by one. The CFD output data for all 13 scenarios is collected as the input to the sensor placement algorithm. For each rack overheating scenario, we set the power dissipation of the overheating rack to 3600 Watts per block, thus 14,400 Watts for a whole rack, which is the same parameter used in [14]. We collect the temperatures and rates of the CARCs' cold air flows using the Tri-Sense digital temperature indicator from Cole-Power [29]. Due to the estimation errors in the CFD software, the computational result from CFD can slightly deviate from the real temperatures in the room. We need to calibrate the data from CFD before using it as input to the sensor placement algorithm. We run the CFD computation for the normal operating server room with a normal workload (server power dissipation) and compare the result from CFD with the real temperature data collected from all the racks, shown in Figure 4. We then derive an offset temperature change  $\delta T$  by the least square error method based on these sample data. The offset value is then compensated back to CFD results for all locations. The calibrated CFD data in Figure 4 is the compensated data at the corresponding locations.

In our analysis, we compare our sensor placement scheme, *CFD+LSP*, with two baseline approaches: *CFD+Proportional* and *Uniformly Random*. The two baselines differ from *CFD+LSP* in that they use different strategies to add sensors to clusters. In *Uniformly Random*, sensors are deployed evenly into clusters. Within a cluster, the placement of sensors is random. In *CFD+Proportional*, sensors are placed in the cluster based on cluster size. A cluster with a larger size gets more sensors. The placement in each cluster is derived from the CSA algorithm with the CFD results as input.

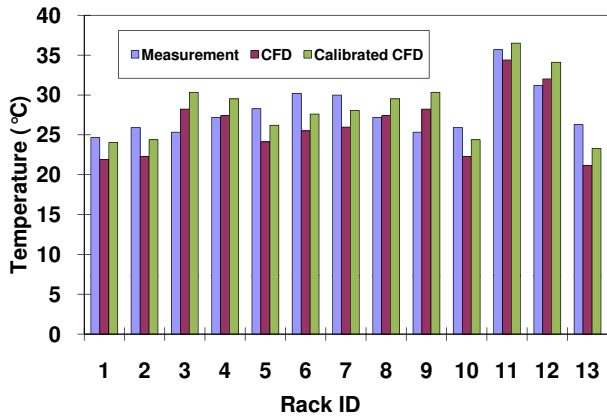


Fig. 4. CFD temperature data calibration.

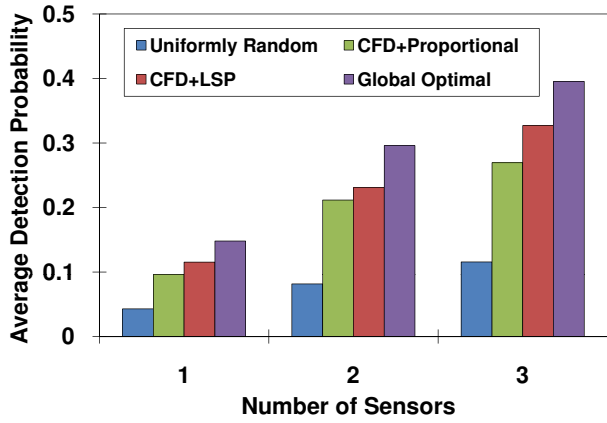


Fig. 5. Average detection probability compared with global optimal solution.

### A. Simulation Results

In this section, we evaluate the different placement schemes with the numerical results. In the first experiment, we explore the average detection probability performance under different sensor numbers. We first show a comparison between different placement performances with the global optimal results. The global optimal results are derived by running CSA solver to solve the placement problem for the entire room without dividing servers into clusters. Because of the tremendously high computational complexity, we are only able to run the optimal solution with, up to three sensors. The results are shown in Figure 5. We can see that *CFD+LSP* shows the closest performance to the global optimal solution with only an approximately 7% difference, while the other two placement schemes are further away from the global optimal solutions. This demonstrates that *CFD+LSP* can better approximate the global optimal solutions.

Figure 6 shows the detection probability performance with more sensors. Compared with *Uniformly Random*, *CFD+LSP* shows a performance improvement of more than 100%. This is because the *CFD+LSP* scheme utilizes the theoretical analysis results from CFD as the basis of the placement algorithm, while *Uniformly Random* does not have any theoretical guarantee when placing the sensor. *CFD+LSP* also shows a 10% incremental increase in the

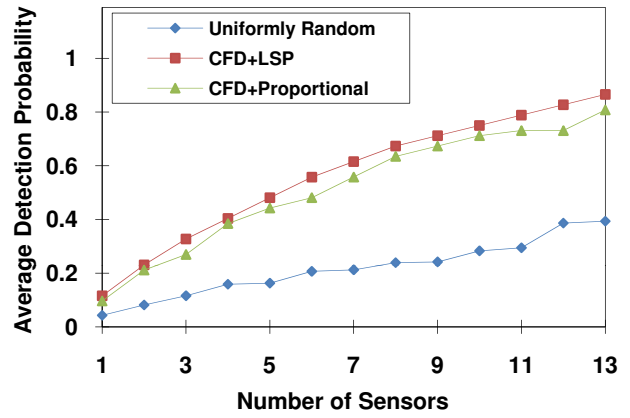


Fig. 6. Average detection probability under different sensor number.

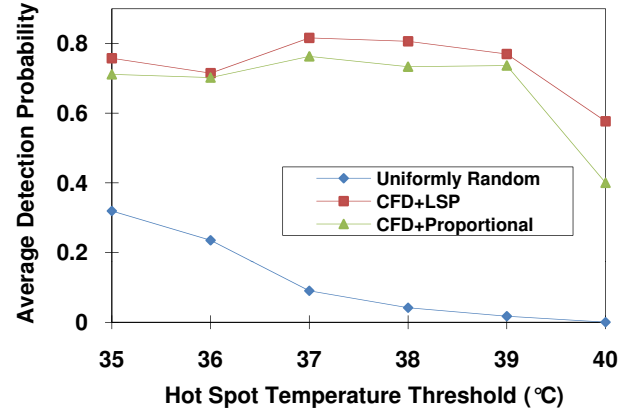


Fig. 7. Average detection probability under different temperature threshold.

detection probability over *CFD+Proportional*. The reason for this improvement is that when choosing a cluster to place each sensor in, we consider the global maximum improvement, while *CFD+Proportional* only considers the optimal placement in each individual cluster.

Figure 7 shows the results of the second experiment, in which we vary the temperature threshold when solving the placement problem. With a higher temperature threshold, it is harder to detect a presence of overheating. Ten sensors are placed by each placement algorithm. From the results we see that *CFD+LSP* shows the best detection probability. It can reach a 60% detection probability even with the highest temperature threshold, while *Uniformly Random* simply does not work when the temperature threshold is high. *CFD+Proportional* shows worse performance than *CFD+LSP*. *CFD+LSP* is the best method since it considers the global maximum when solving the placement solutions.

In the next experiment, we explore the performances of the different placement approaches with different fusion radius settings. The results are shown in Figure 8. With the fusion radius increasing, the average detection probability increases at first, for all three approaches. This is because with a larger fusion radius, more sensors can be involved in the decision making of an overheating location. However, when the fusion radius is larger than 1m, the average detection probability



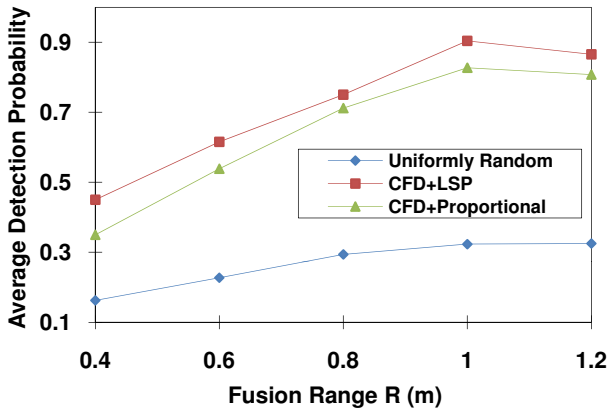


Fig. 8. Average detection probability under different fusion range.

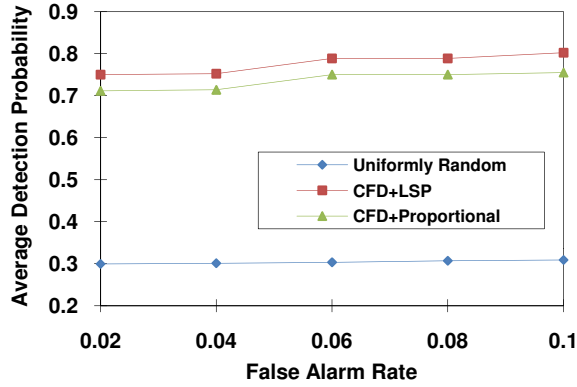


Fig. 9. Average detection probability under different false alarm rate.

begins to decrease. This is because several distant sensors are included in the fusion region of monitored locations when the fusion radius is high, leading to a distortion of the fusion result. Among all three placement schemes, the *CFD+LSP* performs best with an approximately 90% average detection probability when the fusion radius is 1m. *Uniformly Random* performs the worst with only, at most, a 30% average detection probability.

In the the last simulation experiment, we vary the false alarm rate requirements to different values. The results are shown in Figure 9. We see that when the false alarm rate increases from 0.02 to 0.1, the average detection probability increases slightly. For *CFD+LSP*, the increase of the average detection probability is approximately 6%. Among all three sensor placements, *CFD+LSP* performs best, outperforming the *Uniformly Random* placement by about 200% on the average detection probability. It also shows an improvement of more than 10% over the *CFD+Proportional* placement.

### B. Hardware Testbed Results

In this section, we show the experimental results from our testbed in the server room. Due to limited allowed access to the server racks, we establish our testbed in the cluster with 2 racks at the right bottom corner of the server room, as shown in Figure 1. With two racks, the cluster has 8 blocks, *i.e.*, 8 monitored locations in total. Since we are not allowed to change the workloads of the servers to make them

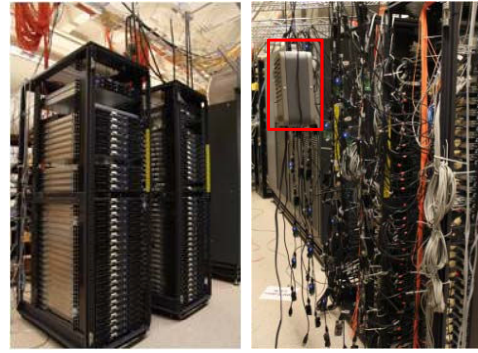


Fig. 10. Front and back sides of the server racks used in hardware experiments (the heater used to emulate overheating server is highlighted).

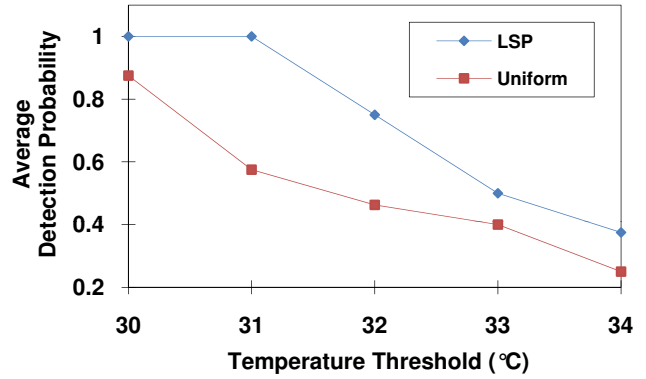


Fig. 11. Average detection probability with different temperature threshold.

overheating, in order to emulate server overheating at each monitored location, we place a small fan heater at the back of the rack to heat up the air around the monitored location, one at a time, as shown in Figure 10. The labeled peak power usage of the heater is 1,500W. Because the heater and a real overheating server still have different thermal behaviors, the CFD results for the entire server room cannot be directly applied. Therefore, we calibrate the CFD model by using Telosb motes to collect the temperature data in the overheating scenarios. The calibrated model is then used to guide our placement algorithm to find the near-optimal sensor placement solutions. We compare our *LSP* scheme with a baseline called *Uniform*, which is similar to the uniform-distance deployment strategy adopted in many real data centers [6][5]. Four sensors are used for the two racks in both methods. In *Uniform*, two sensors are placed on each rack, with one at the 1/4 height and the other at the 3/4 height of the rack.

Figure 11 shows the average detection probability under different overheating temperature thresholds. Each point is the average of 20 repeated experiments. The results show that *LSP* significantly improves detection performance under all thresholds due to its better sensor placement. When the threshold is 31°C, *LSP*'s detection probability is twice that of *Uniform*. Figure 12 shows the temperatures reported by the sensors when the heater is placed at the 8 monitored locations in the case when the temperature threshold is 31°C. We can see that all the overheating scenarios are correctly

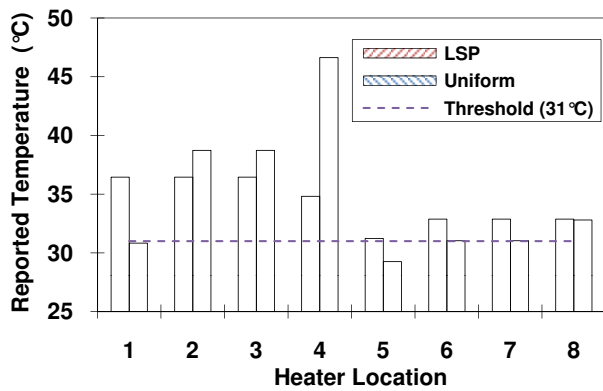


Fig. 12. Reported temperature when the heater is placed in different location (temperature threshold is 31°C).

detected (temperature exceeding the threshold) by *LSP* due to its optimized sensor placement for maximized detection probability in the entire 8 monitored blocks. In contrast, *Uniform* can only detect some of the overheating scenarios with a commonly used uniform sensor deployment strategy.

## VII. CONCLUSIONS AND FUTURE WORK

Efficient thermal monitoring is critical for today's data centers to significantly reduce cooling energy consumption and operating costs. WSN technology has recently been identified as an ideal candidate for data-center thermal monitoring. However, existing solutions adopted by real data centers place sensors in a simplistic way without considering the thermal dynamics in data centers, resulting in an unnecessarily degraded detection probability. In this paper, we have presented a novel sensor placement scheme for hot server detection in data centers based on CFD analysis of thermal dynamics. Our solution features a heuristic algorithm that significantly reduces the computational complexity of finding a near-optimal sensor placement scheme for maximized hot server detection probabilities. Our initial empirical results on a hardware testbed and simulation results both demonstrate that our placement solution outperforms a commonly used uniform placement solution in terms of detection probability.

While our work presents an initial but promising study of intelligent sensor placement in data centers, we plan to make several further improvements in our future work. First, we plan to conduct finer-grained CFD modeling for improved analysis of thermal dynamics. Second, we plan to obtain access to more server racks so that we can conduct larger-scale hardware experiments to demonstrate the efficacy of our solution in detecting hot servers in the entire server room. Finally, we will integrate our sensor placement scheme with well-known thermal control approaches for a complete thermal control loop with both monitoring and actuation in data center cyber-physical systems.

## REFERENCES

[1] "United states environmental protection agency. report to congress on server and data center energy efficiency." 2007. [Online]. Available: [http://www.energystar.gov/ia/partners/prod\\_development/downloads/EPA\\_Datacenter\\_Report\\_Congress\\_Final1.pdf](http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf)

[2] L. Stapleton, "Getting smart about data center cooling." 2006. [Online]. Available: <http://www.hpl.hp.com/news/2006/oct-dec/power.html>

[3] "Wikipedia technical blog," 2010. [Online]. Available: <http://techblog.wikimedia.org/>

[4] Z. Wang, A. McReynolds, C. Felix, C. Bash, C. Hoover, M. Beitelmal, and R. Shih, "Kratos: Automated management of cooling capacity in data centers with adaptive vent tiles," *ASME Conference Proceedings*, vol. 2009, no. 43833, pp. 269–278, 2009. [Online]. Available: <http://link.aip.org/link/abstract/ASMECP/v2009/i43833/p269/s1>

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

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

[7] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling 'cool': temperature-aware workload placement in data centers," in *USENIX*, 2005.

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

[9] X. Wang, X. Wang, X. Fu, G. Xing, and N. Jha, "Flow-based real-time communication in multi-channel wireless sensor networks," in *EWSN*, 2009.

[10] X. Wang, X. Wang, G. Xing, and Y. Yao, "Exploiting overlapping channels for minimum power configuration in real-time sensor networks," in *EWSN*, 2010.

[11] A. J. Shah, V. P. Carey, C. E. Bash, and C. D. Patel, "Exergy-based optimization strategies for multi-component data center thermal management: Part I — analysis," in *IPACK*, 2005.

[12] —, "Exergy-based optimization strategies for multi-component data center thermal management: Part II — application," in *IPACK*, 2005.

[13] J. Moore and J. S. Chase, "Weatherman: Automated, online, and predictive thermal mapping and management for data centers," in *ICAC*, 2006.

[14] C. D. Patel, C. E. Bash, C. Belady, L. Stahl, and D. Sullivan, "Computational fluid dynamics modeling of high compute density data centers to assure system inlet air specifications," in *IPACK*, 2001.

[15] C. D. Patel and A. J. Shah, "Cost model for planning, development and operation of a data center," HP Lab., Tech. Rep., 2005.

[16] J. Choi, Y. Kim, A. Sivasubramaniam, J. Srebric, Q. Wang, and J. Lee, "Modeling and managing thermal profiles of rack-mounted servers with thermostat," in *HPCA*, 2007.

[17] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: maximizing information while minimizing communication cost," in *IPSN*, 2006.

[18] Z. Yuan, R. Tan, G. Xing, C. Lu, Y. Chen, and J. Wang, "Fast sensor placement algorithms for fusion-based target detection," in *RTSS*, 2008.

[19] P. K. Varshney, *Distributed Detection and Data Fusion*. Springer-Verlag New York, Inc., 1996.

[20] T. Clouqueur, K. K. Saluja, and P. Ramanathan, "Fault tolerance in collaborative sensor networks for target detection," *IEEE Transactions on Computers*, vol. 53, pp. 320–333, 2003.

[21] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for target detection," in *WSNA*, 2002.

[22] W. Feller, *An introduction to probability theory and its applications*. John Wiley & Sons, Inc., 1968.

[23] "Crossbow technology, telosb mote." [Online]. Available: <http://www.xbow.com/Products/productdetails.aspx?sid=252>

[24] "CFD Flow Modeling Software and Solutions from Fluent." [Online]. Available: <http://www.fluent.com>

[25] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation, New York, 1980.

[26] B. W. Wah, Y. Chen, and T. Wang, "Simulated annealing with asymptotic convergence for nonlinear constrained optimization," *J. of Global Optimization*, vol. 39, no. 1, pp. 1–37, 2007.

[27] E. H. Isaacs and R. M. Srivastava, *An introduction to applied geostatistics*. Oxford University Press, 1989.

[28] J. Li and A. D. Heap, *A review of spatial interpolation methods for environmental scientists*. Geoscience Australia, Canberra, 2008.

[29] "Cole-palmer: Scientific instruments and lab supplies." [Online]. Available: <http://www.coleparmer.com/index.asp>