Thermal-Aware Task Scheduling for Data centers through Minimizing Heat Recirculation

Qinghui Tang, Sandeep K. S. Gupta and Georgios Varsamopoulos

The IMPACT Laboratory
School of Computing and Informatics
Arizona State University
Tempe, AZ 85287
{qtang,sandeep.gupta,georgios.varsamopoulos}@asu.edu
http://impact.asu.edu/

***Abstract*— The thermal environment of data centers plays a significant role in affecting the energy efficiency and the reliability of data center operation. A dominant problem associated with cooling data centers is the *recirculation* of hot air from the equipment *outlets* to their *inlets*, causing the appearance of *hot spots* and an uneven inlet temperature distribution. Heat is generated due to the execution of tasks, and it varies according to the *power profile* of a task. We are looking into the prospect of assigning the incoming tasks around the data center in such a way so as to make the *inlet* temperatures as *even* as possible; this will allow for considerable cooling power savings. Based on our previous research work on characterizing the heat recirculation in terms of cross-interference coefficients, we propose a task scheduling algorithm for homogeneous data centers, called XInt, that minimizes the inlet temperatures, and leads to minimal heat recirculation and minimal cooling energy cost for data center operation. We verify, through both theoretical formalization and simulation, that minimizing heat recirculation will result in the best cooling energy efficiency. XInt leads to an inlet temperature distribution that is $2^{\circ}C$ to $5^{\circ}C$ lower than other approaches, and achieves about 20%-30% energy savings at moderate data center utilization rates. XInt also consistently achieves the best energy efficiency compared to another recirculation minimized algorithm, MinHR.**

I. Introduction

Increasingly, compute clusters and server farms are housed in data centers that are limited by power and thermal capacity. For a large scale data center, the annual energy cost can be up to millions of dollars, and the cooling cost is at least half of the total energy cost [1]–[3]. Minimizing the energy cost and improving the thermal performance of data centers is one of the key issues toward optimizing computing resources, improving utilization, and maximizing computation capability. In this paper, we explore the potential of reducing the operational cost of data centers by making cooling more efficient (and thus more economical).

As computing devices in a data center emit heat by running tasks, the cooling system must supply cool air to their air inlets that is below their **redline temperature**, i.e. the maximum operational temperature specified by the device manufacturer. Although, ideally, the supplied cool air should be slightly below that threshold, in reality, due to the complex nature of airflow inside data centers, some of the hot air from the outlets of the servers **recirculates** into the inlets of other servers. The recirculated hot air mixes with the supplied cold air and causes all the inlets to experience a rise in temperature at various levels. Thus—and that's how it is in real data centers—the temperature of the air supplied by the cooling system has to be set much lower than the redline temperature, low enough to bring all the inlet temperatures well below the redline threshold.

Lowering the output temperature of the cooling system forces it to operate at a worse **coefficient of performance** (i.e. the ratio of the removed heat over the energy required to do so), which considerably increases the cooling cost. Therefore, one of the reasons of *extra energy cost for cooling data centers is mainly due to the recirculation of hot air* back to the equipment inlets. Reducing the recirculation can be achieved at design time, by improving the physical layout of a data center. Beyond that, recirculation can be further reduced at runtime, by adjusting the data center operation, such as by dynamically assigning tasks in a thermal-aware manner, which is our concentration in this work.

In this paper, we exploit the abstract heat recirculation model, as well as the thermal profiling techniques proposed in our previous work [4] to formulate the problem of minimizing the heat recirculation by appropriately assigning the incoming tasks around the servers. Solving this problem, consequently results in minimal cooling energy cost for the data center's operation. Naive or rule-of-thumb approaches such as assigning tasks to the coolest locations fail to consider the impact of heat recirculation and may lead to worse energy cost in some scenarios. We verify, through both theoretical formalization and simulation, that minimizing heat recirculation will result in the best energy efficiency. This paper's contribution can be summarized as follows:

- Characterization and quantification of heat recirculation as cross interference.
- Mathematical formalization of the problem of minimizing data center energy cost as the problem of minimizing heat recirculation and minimizing maximal inlet temperatures.

- Development of thermal-aware and task-oriented (instead of workload-oriented or power-oriented) scheduling algorithm based on cross interference minimization.

The rest of the paper is organized as follows: Section II provides a discussion on the system model and formulates the problem of minimizing the heat recirculation as . Section

II. System Model and Problem Formulation

A. Preliminaries

A typical data center is laid out with a hot-aisle/cold-aisle arrangement by installing the racks and perforated floor tiles in the raised floor. The air conditioners, normally referred to as Computer Room Air Conditioner (CRAC) or Heating Ventilation Air Conditioner (HVAC), deliver cold air under the elevated floor. In the sequel, this is referred to as *cool air*. The cool air enters the racks from their front side, picks up heat while flowing through these racks, and exits from the rear of the racks. The heated exit air forms hot aisles behind the racks, and is extracted back to the air conditioner intakes, which, in most cases, are positioned above the hot aisles. Each rack consists of several chassis, and each chassis accommodates several computational devices (servers or networking equipment).

In typical High Performance Computing (HPC) scenarios, the servers perform a task for hours or even days. For example, a Spice circuit simulation task of a new VLSI design may run the simulation in parallel on hundreds of servers for days. In a data center with 2000 processors, a task that requires 10% data center capacity means the task size is 200, and 200 processors are required to perform the current task.

In closed environments like data centers, when there is a change in the distribution of power consumption, the temperature distribution reaches a new steady state in about 10 to 20 minutes [5]. Therefore, we assume the data center stays in a certain utilization rate long enough for the temperature distribution under this utilization rate to reach a steady state. This is especially true for HPC data centers where tasks take days to finish.

For simplicity, we assume a **homogeneous** environment. All nodes (chassis) contain the same number of server blades, which have the same power consumption and same computing capability. Also, in the problem formulation to follow, we assume the simple situation of assigning a single, multi-processor task to an idle data center.

In this section we show that there is a dependency between the cooling energy and the placement of a task around a data center. This is achieved by the following steps:

(II-B) First we show the effects of the coefficient of performance to the energy needs of the data center, and provide a relation of the supplied cool air temperature to the power consumed in the data center.

(II-C) We express the inlet temperatures in terms of power consumed, using an abstract heat recirculation model from our previous work [4].

(II-D) We express the inlet temperatures in terms of the task power profile and placement.

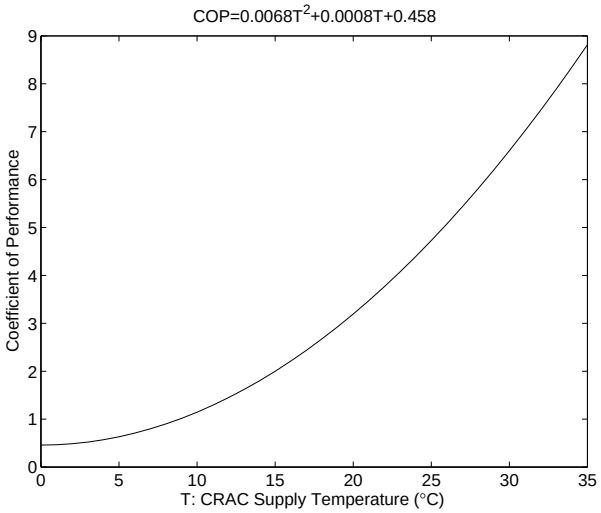

Fig. 1. Coefficient of performance curve for the chilled-water CRAC units at the HP Labs Utility Data Center. (source: [6])

(II-E) Lastly, we express the maximum allowed supplied cool air temperature in terms of the maximum inlet temperature, which is in turn expressed in relation to the task placement.

Overall, we show that we can reduce the total power needs of the data center by solving the problem of minimizing the maximum inlet temperature.

B. Importance of coefficient of performance in total consumed power

The total energy cost, or usage, of data centers is composed of the total computing energy cost—from both computing and networking devices—and the total cooling energy cost. Incidental energy costs such as that for facilities lighting are considered to be of negligible contribution to the total energy cost. The total computing power consumption P_c is presented as:

$$P_c = \sum_{i=1}^{n} P_i.$$

The cooling energy cost can be described as [6]:

$$P_{AC} = \frac{P_c}{CoP(T)}, \tag{1}$$

where CoP is the **coefficient of performance** of the cooling device. CoP characterizes the efficiency of an air conditioner system, it is defined as the ratio of the amount of heat removed by the cooling device to the energy consumed by the cooling device. For example, a ratio of two indicates that to remove heat at the rate of 1000 W, the workload performed by the cooling device is 500 W.

As shown in Figure 1, we use the CoP model used in [6], which is obtained from a water-chilled CRAC unit in HP Utility Data center:

$$CoP(T) = (0.0068T^2 + 0.0008T + 0.458), \tag{2}$$

TABLE I
SYMBOLS AND DEFINITIONS.

Symbol	Definition
n	The number of computing nodes
m	the number of servers (blades) in node i
q	the number of incoming tasks
C_{tot}	the number of servers (blades) the task requires
b	the power consumption of a server at node i running task k
a	idle power consumption of node i's power unit
c_p	Specific heat of air (typical value: $1005J/kg/K$)
ρ	Density of air (typical value: $1.19kg/m^3$)
C_i	The number of tasks assigned to server i
P_i	Power consumption of node i
T_{sup}	air temperature as supplied from the cooling unit
T_{red}	manufacturer's redline temperature (typical value 25 °C)
T^i_{in}	Inlet air temperature of node i
T^i_{out}	Outlet air temperature of node i
f_i	Flow rate of node i (typical value 520 CFM = 0.2454 m^3/s)
Q_i	Heat dissipation of node i
K_i	the thermodynamic constant, short for $\rho f_i c_p$
$\mathbf{A}$	The heat cross-interference coefficient matrix
$\mathbf{K}$	thermodynamic constant matrix $\mathbf{K} = \text{diag}(K_i), i = 0 \ldots n$
$\mathbf{D}$	distribution matrix, concise for $[(\mathbf{K} - \mathbf{A}^\intercal \mathbf{K})^{-1} - \mathbf{K}^{-1}]$
$\overrightarrow{\mathbf{P}}$	the vector $\{P_i\}_n$
$\overrightarrow{\mathbf{T}}_{in}$	the vector $\{T^i_{in}\}_n$
$\overrightarrow{\mathbf{T}}_{out}$	the vector $\{T^i_{out}\}_n$

where T is the temperature of the supplied cold air. Note that the change of CoP is not linear and normally increases with the supplied air temperature. We can observe that operating the cooling system at a higher temperature is saving energy. Intuitively, to provide colder air, the cooling device has to work harder and consume more energy to remove more heat from the supplied cold air.

Therefore, we can minimize P_{AC} by maximizing the supplied cold air temperature T while satisfying the constraints of redline threshold.

The total energy consumption for operating a data center is defined as:

$$P_{Total} = P_{AC} + P_c, \tag{3}$$

$$= \left(1 + \frac{1}{CoP(T_{sup})}\right) \sum_{i=1}^{n} G_i(C_i), \tag{4}$$

C. Power consumed in computation affects the inlet temperatures

A data center is abstracted to consist of n **nodes** (chassis). Each node i consists of m **servers** (blades). Each node i draws air over the node i with inlet temperature T^i_{in}, and dissipates hotter air with average outlet temperature T^i_{out}. The outlet temperature of a node comes from the combined activity of the servers in that node, while the inlet temperature comes from the combination of cool air supplied from the air-conditioning and hot air recirculated from the node outlets. Contemporary data centers are cooled by using conventional air-cooled technology.

According to the *law of energy conservation*, the amount of heat or energy carried by an air flow per unit time is:

$$Q = \rho f c_p T,$$

where ρ is the air density, f is the air flow rate, c_p is the specific heat of air, and T is the air temperature. Due to the unique location of each node in the data center, we assume that the air flow rate differs for each node. We will denote as f_i the air flow of the node i.

Considering that, in a steady state, the power drawn by a computing device is dissipated as heat, the relationship between power consumption of a node and the inlet/outlet temperature can be written as:

$$P_i = \rho f_i c_p \left(T^i_{out} - T^i_{in}\right) \ ,$$
$$T^i_{out} = T^i_{in} + K_i P_i, \text{ where } K_i = \rho f_i c_p \ .$$

In other words, the power consumption of node i will cause air passing through the node i to experience an energy increase of P_i, and a temperature rise from T^i_{in} to T^i_{out}.

According to the abstract heat model of the data center, as described in previous work [4], the recirculation of heat can be described by a cross-interference coefficient matrix $\mathbf{A}_{n\times n} = \{\alpha_{ij}\}$, which denotes how much of its heat each node contributes to every other node. That is, the matrix element α_{ij} denotes that α_{ij} heat output from node i recirculates into node j. Note that $\sum_{i=0}^{n} \alpha_{ij} \leq 1$, but $\sum_{j=0}^{n} \alpha_{ij} \lneqq 1$.

If the thermodynamic constants K_i are organized into a diagonal matrix $\mathbf{K}_{n\times n} = \text{diag}(K_1, K_2, \ldots K_n)$, it is shown [4] that the vector of inlet temperatures $\overrightarrow{\mathbf{T}}_{in}$ can be expressed as:

$$\overrightarrow{\mathbf{T}}_{in} = \overrightarrow{\mathbf{T}}_{sup} + \left[(\mathbf{K} - \mathbf{A}^\intercal \mathbf{K})^{-1} - \mathbf{K}^{-1}\right] \overrightarrow{\mathbf{P}},$$

For brevity, we will denote $\mathbf{D} = \left[(\mathbf{K} - \mathbf{A}^\intercal \mathbf{K})^{-1} - \mathbf{K}^{-1}\right]$, so the equation is formulated as

$$\overrightarrow{\mathbf{T}}_{in} = \overrightarrow{\mathbf{T}}_{sup} + \mathbf{D}\overrightarrow{\mathbf{P}}, \tag{5}$$

which means that each inlet temperature is above the supply temperature by the excess heat from recirculation. We can see that *the row in the product* $\mathbf{D}\overrightarrow{\mathbf{P}}$ *with maximum value determines the row in* $\overrightarrow{\mathbf{T}}_{in}$ *with the maximum value*.

The next subsection links the power dissipation to the served tasks by introducing a **power profile** that maps a task to its power needs with respect to the running node.

D. Task placement affects the inlet temperatures

As mentioned in the preliminaries, the data center is given a task of "size" C_{tot} to run. For simplicity, we assume that the size of the task means the number of processors required. A task of size 20 means the task requires 20 processors. Each node contains m processors. A scheduler dispatches the task to n nodes, each node will run a "sub-task' with the size of C_i. Of course the scheduling results should satisfy the constraints that

$$\sum_{i=1}^{n} C_i - C_{tot} = 0, \text{ and } C_i \leqslant m.$$

An incoming task consists of a sequence of machine-code instructions. Each instruction has a different power need, according to the specifications of the manufacturer. The combined

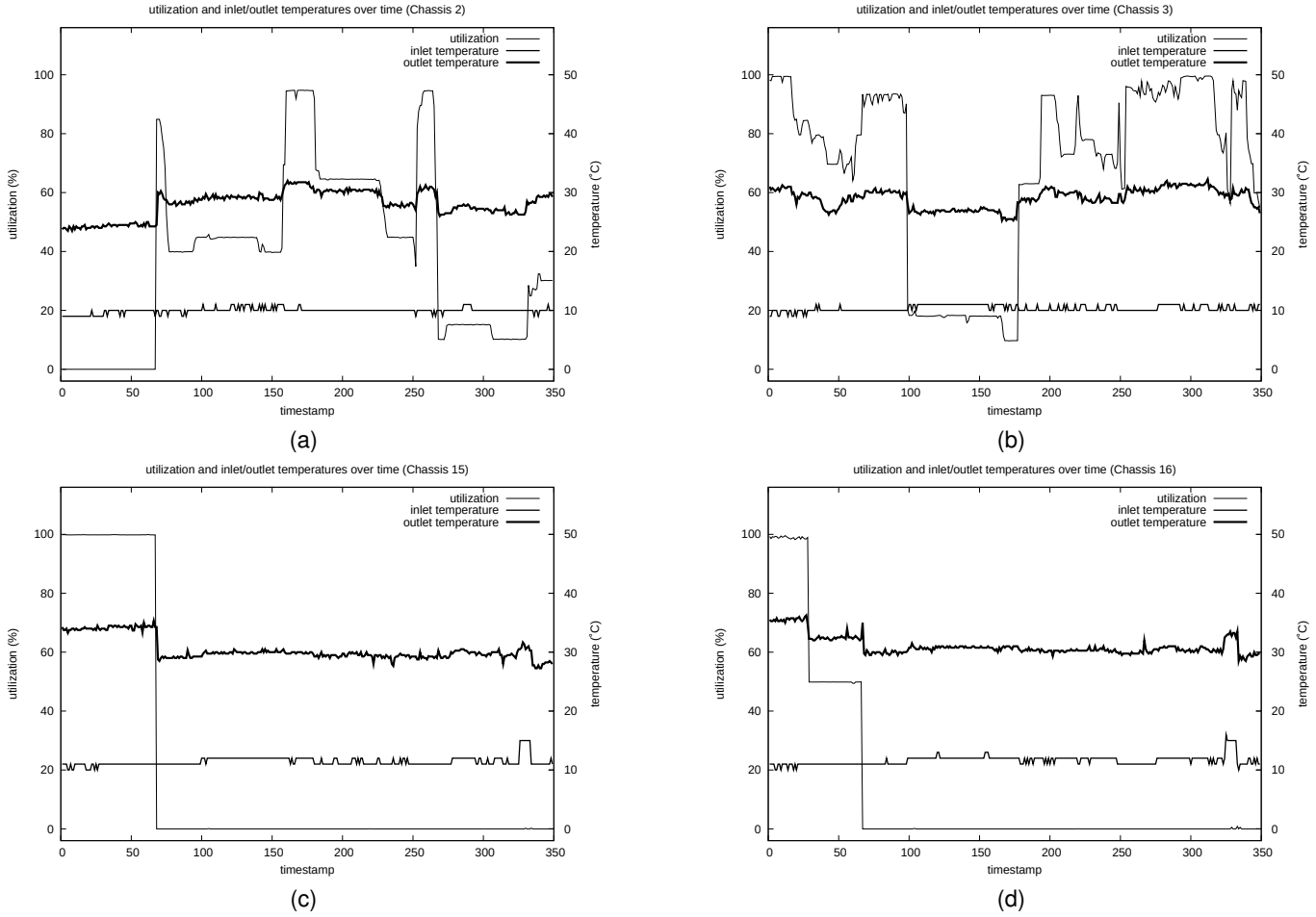

Fig. 2. Example graphs of utilization, inlet and outlet temperatures over time, for four selected chassis. All graphs show the direct effect of utilization on outlet temperature, while the lower ones also show the effect of inlet temperature to the outlet one (notice the temperature "bumps" around timestamp 330).

execution of those instructions determine the power profile of a machine. Although the power consumption changes with every instruction executed, the analysis in this paper assumes that each task has a power consumption constant with respect to time.

Moreover, the organization of our data center abstraction conforms with the modern organization of data centers into chassis and blades. A blade does not have its own power unit; it relies on the power unit of the chassis it is in. If the node expends a power at idle, and each blade expends b power when running a specific task, the power consumption of a chassis with m blades is modeled as:

$$P = a + mb.$$

This is a *linear* power model with respect to CPU utilization. Linear power models with respect to CPU utilization are quite accurate given their simplicity [4], [7], [8]. There is also a linear dependency of outlet temperature (heat dissipation) to the power consumed. Fig. 2 shows selected samples of utilization, inlet and outlet temperature graphs from blade server chassis at the ASU Fulton High Performance Computing Center; in the graphs, there is a strong dependency of the outlet temperature to the utilization, which gives a basis to the utilization-based power models.

Since a node contains many servers, a node may run many tasks. In a high-performance computing data center, tasks usually require many processors (servers) to run on. Therefore, when a task runs on C_i servers on a node i, then the power needs of that task is $C_i b$. From the above, when a node i runs a task on C_i blades, the power consumption of that node is

$$P_i = a + C_i b \tag{6}$$

Taking it a step further, we can construct a vector $\overrightarrow{\mathbf{C}}$ of the values C_i across the data center, the power vector $\overrightarrow{\mathbf{P}}$ is expressed as:

$$\overrightarrow{\mathbf{P}} = \begin{bmatrix} a & a & \dots & a \end{bmatrix}^{\mathsf{T}} + \overrightarrow{\mathbf{C}} b, \tag{7}$$

Applying Eq. 7 to Eq. 5, we get that:

$$\mathbf{T}_{in} = \mathbf{T}_{sup} + \mathbf{D} \begin{bmatrix} a & a & \dots & a \end{bmatrix}^{\mathsf{T}} + \mathbf{D}\overrightarrow{\mathbf{C}} b \tag{8}$$

In the above equation, $\mathbf{D}$ is not adjustable. On the other hand, vector $\overrightarrow{\mathbf{C}}$ is. Considering the analysis in Subsectionsec:power-inlet, it is now evident that *the inlet temperature vector depends on the placement of jobs around*

the data center. The next subsection provides the dependency of the supplied cool air temperature on the maximum inlet temperature.

E. Inlet temperatures versus supplied cool air temperature

Suppose that, from the Eq. 8 the difference between the peak inlet temperature and the red-line temperature T_{red} is

$$\Delta = T_{red} - \max_i\{T_{in}^i\}.$$

As mentioned earlier, cooling systems supply cool air at a temperature well below the redline value. We can adjust the supplied cold air temperature to a higher value T'_{sup}:

$$T'_{sup} = T_{sup} - \Delta = T_{sup} - \max_i\{T_{in}^i\} + T_{red},$$

which is the point at which one of the inlet temperatures reach the redline.

Thus, maximizing the supplied cold air temperature T'_{sup} is equal to the problem of minimizing $\max_i\{T_{in}^i\}$ given T_{sup}. In addition, assuming no heat transfer from any other sources, e.g., fans, walls, etc, the only reason that a server may have a higher inlet temperature than that of others is because it suffers from more heat recirculation than others. Therefore, minimization of the peak inlet temperature is equal to minimizing recirculation.

The optimization problem of Eq. 8 can be transformed as

$$\begin{aligned} &\text{minimize}(\max_i\{T_{in}^i\}) \qquad (9)\\ &st: C_{tot} - \sum_{j=1}^{n} C_j = 0\\ &\quad : \overrightarrow{\mathbf{T}}_{in} = \mathbf{T}_{sup} + \mathbf{D}\begin{bmatrix} a & a & \dots & a \end{bmatrix}^{\mathsf{T}} + \mathbf{D}\overrightarrow{\mathbf{C}}b\\ &\quad : C_j \geqslant 0, j = 1 \dots n,\\ &\quad : m - C_j \geqslant 0, j = 1 \dots n. \end{aligned}$$

The intuitive meaning of the optimization problem is how to divide the total task C_{tot} into a task vector $\overrightarrow{\mathbf{C}} = \{C_1, C_2, \dots C_n\}$ to achieve the minimal maximum inlet temperature.

F. Summary

In this section we showed that the computational power consumption is irrespective of the task placement. However, the task placement affects the heat recirculation. In short, the following problems are equivalent:

- Minimization of the cooling energy cost,
- Minimization of the heat recirculation, and
- Minimization of the peak inlet temperature given a reference supplied cold temperature T_{sup}.

To minimize the cooling energy cost of a data center, we are to minimize the heat recirculation, which is equivalent to reducing the maximal inlet temperature of all the server nodes. In the following section, we present **XInt**, a recirculation-minimizing scheduling algorithm for minimizing the peak inlet temperature.

Algorithm 1 XInt: Recirculation-minimizing algorithm

```
1: procedure XINT(C_tot, n, m, T_sup, D, a, b)
2:     CurGen ← a pool of solutions [C_tot/n, C_tot/n, ... C_tot/n] (Eq. 10)
3:     for i ← 1 to MaxGen do
4:         SelSubs ← a randomly selected subset from CurGen
5:         MuSubs ← mutation of solutions in SelSubs
6:         MaSubs ← mating of solutions in SelSubs
7:         Apply the fitness function (Eq. 11) on CurGen, MuSubs, MaSubs
8:         CurGen ← all fit solutions, i.e. ones with low peak inlet temperature
9:     end for
10:    FinalSolution ← the solution within CurGen with best fitness
11: end procedure
```

III. The algorithm XInt

In the previous section, we formulated the problem of minimizing the peak (maximum) inlet temperature as an ILP. We now briefly describe how we use a *genetic algorithm* (GA) optimization approach to find a near-optimal scheduling result. That is, algorithm XInt is a stochastic optimization algorithm that simulates the processes of evolution in nature [9]: mutation and survival of the fittest.

For thermal-aware scheduling, we start with generating the initial populations by equally dividing the total task among all server nodes. Suppose each generation has 100 solutions, then all of them are in the form of

$$\begin{aligned} S_1 &= [\frac{C_{tot}}{n}, \frac{C_{tot}}{n}, \dots \frac{C_{tot}}{n}]\\ S_2 &= [\frac{C_{tot}}{n}, \frac{C_{tot}}{n}, \dots \frac{C_{tot}}{n}]\\ &\vdots \qquad (10)\\ S_{100} &= [\frac{C_{tot}}{n}, \frac{C_{tot}}{n}, \dots \frac{C_{tot}}{n}] \end{aligned}$$

During the mating process, we randomly select several pairs of solutions, exchange a subset of two task assignments and obtain two new solutions. During the mutation process, we randomly select one solution and change its task distribution. For example, we move some task from one node to another node and get a new task assignment solution.

We then conduct some verification and necessary modifications to make sure that the newly generated solutions obtained the mutation and mating process are legitimate ones, such as the sum of all tasks should be equal to C_{tot} or the number of tasks of each node should be less than or equal to m.

The task assignment vector itself can be treated as the chromosomes of each solution. Then, using Eq. 8 we can obtain the fitness F_j of the solution S_j as

$$F_j = \max_i \left\{T_{in}^i\right\}. \qquad (11)$$

The solutions of the current population and the newly generated solutions go through a probability-based *roulette wheel* selection process [9]. Solutions with good temperature distributions would be selected with higher probability and survive into next round of evolution.

IV. Other Approaches

We present the following approaches to the task assignment problem, which will be used to compare the performance of the XInt algorithm.

A. Naive Scheduling Algorithms

First we briefly review three naive thermal-aware scheduling algorithms presented in our previous work [10]. They are based on observation and intuition instead of taking into account the heat recirculation phenomenon.

Uniform Outlet Profile (UOP): This scheme is similar to the OnePassAnalog algorithm presented in [6]. Based on the inlet temperature of each computing node, the algorithm will assign more tasks to nodes with low inlet temperatures, and fewer tasks to nodes with high inlet temperatures. The objective is to achieve a uniform outlet temperature distribution.

Minimal Computing Energy (MCE): MCE minimizes the number of powered-on chassis and processors to concentrate computing energy costs on those active servers and processors and turns all other idle processors or blades off. For the homogeneous data center used in our study, the computing nodes with the lowest inlet temperature will be assigned tasks first. This approach is the same as traditional thermal engineer method: placing load close to the floor vent (coolest temperature locations).

Uniform Task (UT): With this scheme, all nodes are assigned the same amount of tasks: $C_i = C_{tot}/n, \forall i$.

B. MinHR: Recirculation Minimized Algorithm

Our work on XInt was partially motivated by the work in [6], where MinHR, a heat recirculation minimization approach was proposed. MinHR is based on calculating the *Heat Recirculation Factor* (HRF) for each *pod*, i.e. a chassis or a rack, then assigning tasks according to the ratio of one pod's HRF to the sum of all HRFs. In other words, MinHR assigns fewer tasks to pods that cause higher recirculation, which has the same underlying principle as XInt.

1) MinHR problem description: Given a reference workload that generates a given heat load Q_{ref}, which is the sum power consumption (heat dissipation) of all nodes at the reference state, the recirculated heat within this reference scenario is defined in [6] as:

$$Q_{ref} = \sum_{i=1}^{n} \rho f_i c_p (T_{out}^i - T_{in}^i), \tag{12}$$

$$\delta Q_{ref} = \sum_{i=1}^{n} \rho f_i c_p (T_{in}^i - T_{sup}), \tag{13}$$

and Heat Recirculation Factor (HRF) is defined as [6]:

$$HRF_j = \frac{\text{the change in total heat dissipation}}{\text{the change in total heat recirculation}} = \frac{Q_j - Q_{ref}}{\delta Q_j - \delta Q_{ref}},$$

where Q_j and δQ_j are the total amount of heat and the recirculated heat for j^{th} profiling scenario (or j^{th} pod), respectively. HRF can be obtained through a series of profiling steps [6]. Then the total workload is distributed among pods according to the percentage of HRF [6]:

$$P_j = \frac{HRF_j}{\sum_{j=1}^{n} HRF_j} P_{total}. \tag{14}$$

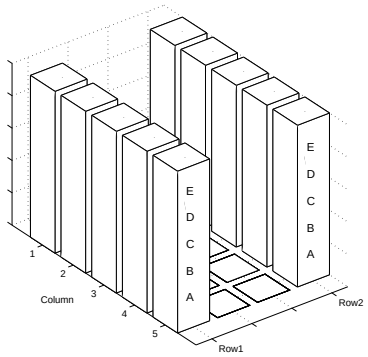

Fig. 3. Two Row data center used in our simulation study, each rack has 5 blade server chassis, marked from bottom to top as A, B, C , D and E.

A small HRF value indicates a pod is a strong recirculation contributor, and the pod will be allocated with less workload.

To compare MinHR with XInt, we propose a Modified *MinHR* (MinHR-m): Since the original MinHR is not designed for placing task but placing power budget, we first obtain task placement result of XInt, then use the corresponding computing energy cost of XInt as the power budget of MinHR-m, thus the two algorithms have the same amount of computing energy and fair comparison is established. If the assigned power consumption is larger than the peak power consumption or less than the idle power consumption, we set these servers' power consumption as the peak value or as the idle value; then we mark the status of the peak value nodes as task-allocated and remove them from the candidate pods. Next, we deduct the allocated power from the total power workload, and repeat the assignment among the remaining candidate pods using the same HRF based proportional assignment. The procedure is repeated until all pods receive a power workload that is within the operational range.

V. Results

A. Simulation Setup

We used Flovent [11], a CFD simulation software to conduct simulations to obtain the thermal distribution for the various scheduling algorithms. We simulated a small scale data center with physical dimensions $9.6m \times 8.4m \times 3.6m$ (see Figure 3), which has two rows of industry standard $42U$ racks arranged in a typical cold aisle and hot aisle layout. The cold air is supplied by one computer room air conditioner, with the flow rate $8m^3/s$. The cold air rises from raised floor plenum through vent tiles, and exhausted hot air returns to the air conditioner through ceiling vent tiles. There are 10 racks and each rack is equipped with 5 chassis (marked from bottom to top as A, B, C, D and E). The maximum computing capacity in the unit of number of processor is 1000. The total power consumption of the whole data center would be 232KWatts at full utilization rate.

Figure 4 shows the inlet temperature distribution when all the servers are idle. Obviously, the chassis located at the lower

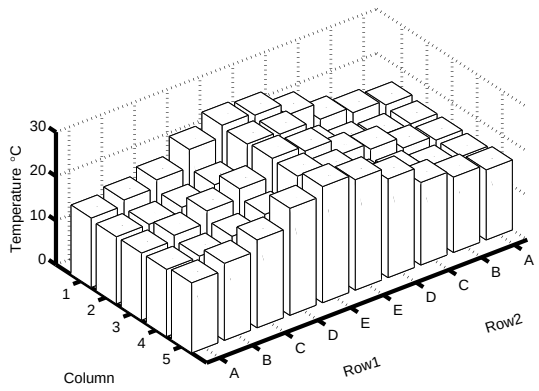

Fig. 4. Inlet temperature distribution at **idle**: chassis locate at the lower part of the rack obtain plenty of cold air from floor vent and have a low inlet temperatures

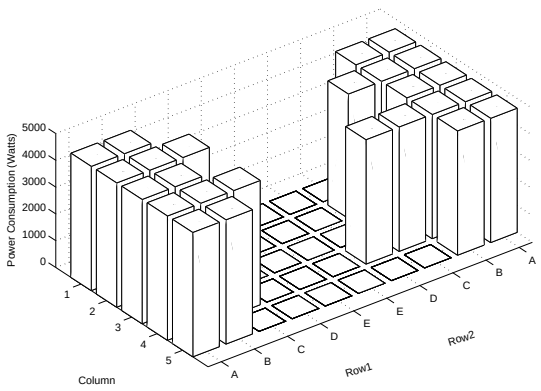

Fig. 5. Power Consumption Distribution of MCE: MCE assign tasks to the nodes located at the lower part of racks which have relatively low inlet temperatures.

part of the rack (A and B) obtain plenty of cold air from the floor vents and have a lower inlet temperature, where chassis located at the upper part (E) of the rack experience a highest inlet temperature due to the insufficient supply of cold air.

B. Comparison with respect to temperatures

First we would like to observe whether different algorithm have any impact on temperature distribution. Figures 5 and 6 compare the difference among the power consumption distributions of three different scheduling results, with the total data center utilization rate at 50%. Implicitly, it also compares the difference among task assignment results. Figure 5 shows that MCE assigns tasks to nodes with lowest inlet temperature, which are located at the lower part of racks in our studied model.

In Figure 6, we observe that XInt assigns tasks to nodes with minimal recirculation: in this case, nodes locating at the upper part obtain the task, since their outlets are close to ceiling vents and cause less recirculation.

Figure 7 and 8 show resulting temperature distributions. The peak inlet temperatures for XInt and MCE are 25.6°C and 30.5°C, respectively. The 5°C temperature difference will result in significant difference in demand for the cooling capability.

Figure 9 compares the maximum and standard deviation of inlet temperatures under four different algorithms. XInt always has the minimal maximum temperature. In addition, it possesses the minimal temperature standard deviation, which indicates that the inlet temperatures of nodes under this scheme are distributed in a relatively smaller range, or they are more evenly distributed when compared with other algorithms. Therefore, we believe the standard deviation of the inlet temperature distribution is the best metric to quantify the degree of recirculation.

C. Comparing with respect to cooling cost

Figure 10 shows cooling cost comparison for the four aforementioned algorithms. We observe that XInt consistently has the minimal cooling cost. At 50% utilization rate, XInt

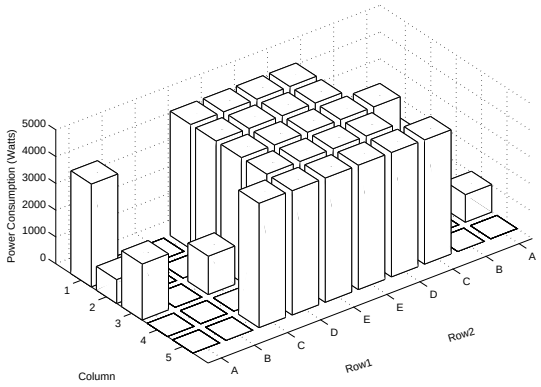

Fig. 6. Power Consumption Distribution of XInt: XInt assigns tasks to nodes who lead to minimal recirculation.

can save 24% to 35% energy cost compared with UT and UOP. In addition, the performance of MCE has the worst energy efficiency for most of the utilization rates. Figure 10 also shows the theoretical optimal lower bound for the cooling cost. The optimal scenario assumes no existence of heat recirculation, and the supplied cold air and all inlet temperatures are redline temperature 25°C.

Figure 11 compares the cooling energy costs of our algorithms with MinHR-m. The energy efficiency of MinHR-m outperforms all the naive algorithms, but is slightly lower than XInt at various utilization rates.

D. Discussion

Intuitively, MCE and the traditional thermal engineer approach provide a reasonably good thermal environment, since they place workload (power consumption) to the locations with lowest temperatures. However, this approach ignores the fact that the nodes with low inlet temperature can be a significant recirculation contributor, and would cause the inlet temperatures of other nodes to rise. The victims of recirculation will demand extra cooling capability and increase the overall data center cooling cost.

XInt persistently achieves the best thermal performance, because it assigns task to nodes that only cause minimal recirculation. Consequently, minimizing recirculation equals

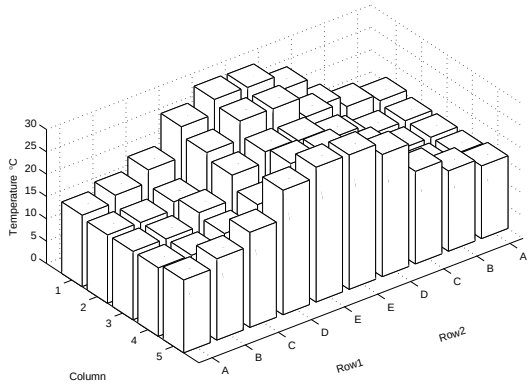

Fig. 7. Inlet temperature distribution of MCE: peak temperature is 30.5°*C*.

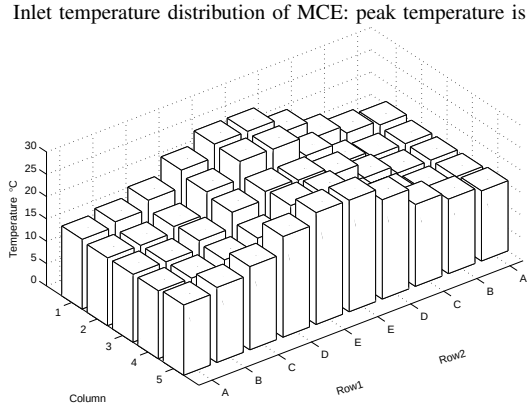

Fig. 8. Inlet temperature distribution of XInt: peak temperature is 25.6°*C*.

to minimizing maximum inlet temperature. Hence, assigning tasks to the nodes with lowest inlet temperature does not necessarily results in good temperature distribution. We have to assign task based on global information, more specifically, based on the global recirculation information.

The performance difference between XInt and MinHR-m can be explained as follows. First, the goal of original MinHR is to minimize the total amount of recirculated heat. Therefore we also need to distribute the recirculated heat among all server nodes as evenly as possible and minimize the peak inlet temperature.

Secondly, the metric HRF used in MinHR-m characterizes an aggregated effect of recirculation (the total amount of recirculation from one node to all the other nodes), it does not show where the recirculated heat goes; whereas our cross interference matrix **A** shows the multi-point to multi-point recirculation among all server nodes (the amount of recirculation from any single node to another single node).

Thirdly, MinHR-m distributes power consumption relative to the ratio of heat produced to heat recirculated. Again, this is based on observation and intuition, and cannot mathematically lead to the optimal solution. Instead, our work *mathematically formalized* the minimizing recirculation problem as the objective function of Eq. 9. That is why MinHR-m has a good performance, but not as consistent as XInt.

The performance of MinHR-m seems close to XInt, however that is based on the prerequisite that XInt helps it obtain the

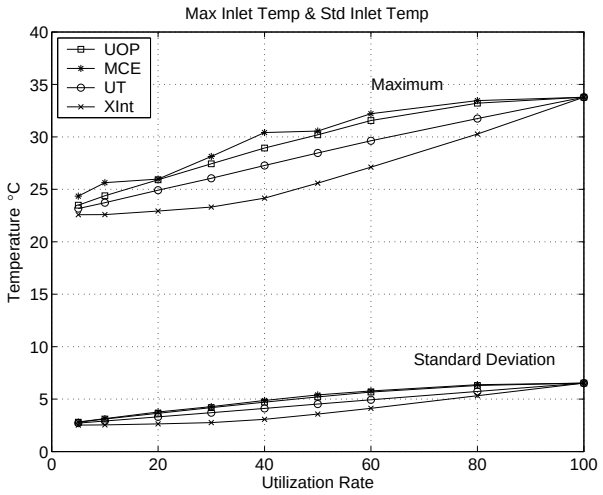

Fig. 9. The maximum and standard deviation of inlet temperature: XInt persistently has the lowest maximum indicates it is the most energy efficient one. XInt has the smallest standard deviation indicates it minimizes and balances the recirculated heat.

total power budget, which is a power budget value that minimizes recirculation due to the nature of XInt algorithm. Even so, they could not outperform XInt because XInt distributes power consumption in a way that minimizes the peak inlet temperature.

Finally, and most importantly, XInt is a *task-oriented placement* algorithm whereas MinHR-m is *power-oriented workload placement* algorithm which has no capacity of placing tasks per se. In reality, a data center administrator does not receive the straight workload in terms of Watts but computes tasks in terms of how many resources/processors are required. For example, it is not the case that an administrator receives a task submission indicating it needs 100kW to run the task and figures out how to distribute the power consumption among server nodes. Instead, an administrator receives an incoming task that may need 300 servers and the problem is which 300 servers to select. Therefore, we believe our task-oriented approach is more applicable in data centers.

E. Evaluation using Heat Indexes

In [12], researchers at HP Labs defined **Supply Heat Index (SHI)** and **Return Heat Index (RHI)** to characterize the energy efficiency of data center cooling systems. SHI is defined as

$$SHI = \frac{\text{Enthalpy rise due to infiltration in cold aisles}}{\text{Total enthalpy rise at the rack exhausts}}$$

and RHI is defined as $1 - SHI$. These dimensionless metrics try to capture and quantify the "badness" of heat recirculation into one scalar value. The lower the SHI value, the better the energy efficiency, since the percentage of recirculated heat is smaller.

Figure 12 shows the measured SHI of the simulated scheduling algorithms. This observation is consistent with the ones from Figure 10: the ranking of SHI at different utilization

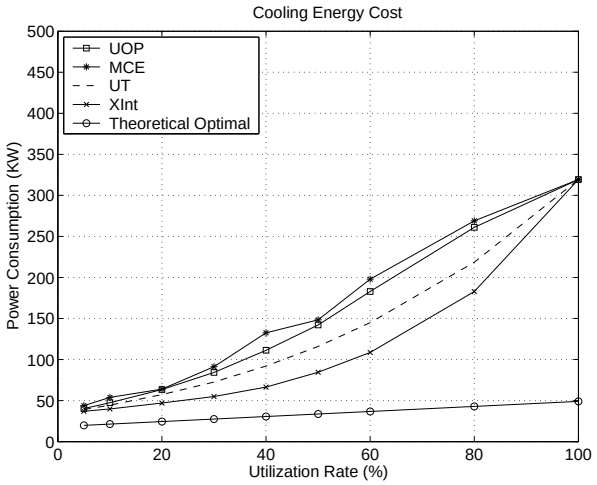

Fig. 10. XInt possesses the minimal cooling energy cost, MCE's energy efficiency is the worst one.

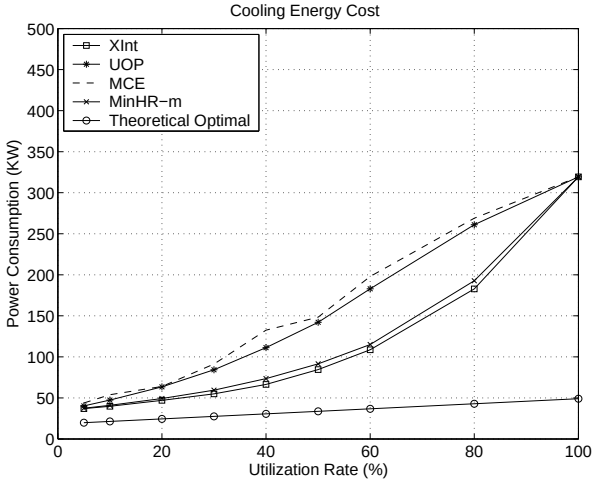

Fig. 11. MinHR has very close performance as XInt, but not as good as XInt

point matches the ranking of energy efficiency. The figure also shows that XInt has the lowest SHI. It is also interesting that MCE, which expresses the idea of placing jobs at the lowest, coolest rows, has the biggest SHI value since it assigns task to recirculation contributors, causing the worst recirculation.

VI. Related Work

There exist *two steps* toward improving the thermal management at the data center level. The *first step* is from the infrastructure design and planning perspective: Data center design and analysis [13], [14] has become increasingly sophisticated, involving CFD modeling in the design phase and increased deployment of temperature sensors (ambient sensors and on-board sensors) and supplementary cooling systems for temperature control and monitoring [15]. The *second* step, which is the focus of this work, is to improve and optimize the thermal performance, especially the temperature distribution, during the operation of a data center. Our contributions follow in the second category

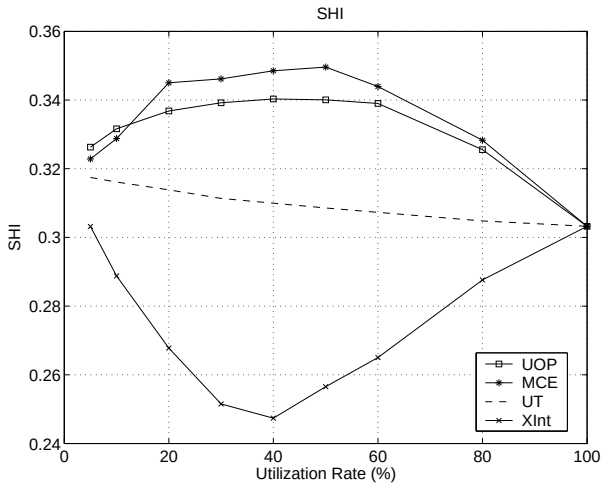

Fig. 12. Supply Heat Index with CareAll policy: the result is consistent with Figure 10, XInt has the minimal SHI since it is the recirculation minimized algorithm.

A. Improving the computation power efficiency

At **chip level**, the work of Multi-core Thermal Management [16] tries to achieve thermal management through changing voltage or migrating process among multiple cores. The power-aware distributed computing for scientific applications project [17] is focusing on improving energy efficiency of large distributed and parallel computing systems by dynamically changing processor voltage without significantly affecting the system performance. Another similar work [18] discussed the problem of minimizing execution time while satisfying energy constraints and time constraints based on a voltage and frequency scalable cluster.

At **chassis level**, after observing the underutilized pattern of typical blade server applications, Ranganathan *et. al.* [19] proposed dynamically redistributing the power budget to avoid inefficient over-provisioning in the cooling and power delivery. They suggested using some typical power control mechanisms such as voltage and frequency scaling.

At **data center level**, some research has also been conducted to reduce computing energy cost of computer systems [20], [21].

B. Improving the cooling power efficiency

Researchers at HP Labs and Duke University have published a series of work [6] [22] [23] [24] and [25] on smart cooling techniques for data centers. They have developed online measurement and control techniques to improve energy-efficiency of data centers.

MinHR [6] is a heat recirculation minimizing algorithm based on calculating the *Heat Recirculation Factor* (HRF) for each *pod*, i.e. usually a chassis or a rack, it assigns fewer tasks to pods that cause higher recirculation, while assigning more tasks to pods that cause less recirculation. Basically, it is a **power-oriented workload placement** algorithm instead of a **task placement** algorithm.

OnePassAnalog and Zone Based Discretization (ZBD) are proposed in work [6] to intuitively assign tasks inversely proportional to the server's inlet temperature. We believe they are similar to the MCE, an intuition based algorithm that cannot guarantee the best energy efficiency.

VII. Conclusions

To improve energy efficiency and reliability of data center operation, thermal-aware workload replacement or scheduling has been studied to improve temperature distribution within data center. Based on our previous research work on characterizing heat recirculation of data center as cross interference coefficient, we propose a task scheduling algorithm, XInt, that leads to minimal heat recirculation and a more evenly distributed temperature distribution, which consequently results in minimal cooling energy cost of data center operation.

We compare the energy efficiency between XInt and other algorithms; XInt consistently achieves the best energy efficiency and saves 20%-30% of energy cost at a moderate data center utilization rate. XInt also outperforms another recirculation-minimizing algorithm named MinHR. We show that the standard deviation of inlet temperature is a better metric to quantify the degree of recirculation inside data center.

VIII. Acknowledgments

We would like to thank Dan Stanzione for granting access to the ASU Fulton HPC Facility, Tridib Mukherjee for performing power measurements, Michael Jonas for data gathering. Parts of this work have been funded by SFAz, Intel Corporation and NSF (CNS#0649868).

References

[1] C. D. Patel, C. E. Bash, R. K. Sharma, A. Beitelmal, and R. J. Friedrich, "Smart cooling of datacenters," in *Proceedings of (IPACK)03 C The PacificRim/ASME International Electronics Packaging Technical Conference and Exhibition*, Kauai, HI, July 2003.

[2] R. F. Sullivan, "Alternating cold and hot aisles provides more reliable cooling for server farms," White Paper, Uptime Institute, 2000.

[3] R. Sawyer, "Calculating total power requirements for data centers," White Paper, American Power Conversion, 2004.

[4] Q. Tang, T. Mukherjee, S. K. S. Gupta, and P. Cayton, "Sensor-based fast thermal evaluation model for energy efcient high-performance datacenters," in *Int'l Conf. Intelligent Sensing & Info. Proc. (ICISIP2006)*, Dec 2006.

[5] J. Moore, R. Sharma, R. Shih, J. Chase, C. Patel, and P. Ranganathan, "Going beyond CPUs: The potential of temperature-aware data center architectures," in *First Workshop on Temperature-Aware Computer Systems*, June 2004.

[6] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling "cool": Temperature-aware resource assignment in data centers," in *2005 Usenix Annual Technical Conference*, April 2005.

[7] T. Heath, A. P. Centeno, P. George, L. Ramos, and Y. Jaluria, "Mercury and freon: temperature emulation and management for server systems," in *ASPLOS-XII: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*. New York, NY, USA: ACM Press, 2006, pp. 106–116.

[8] P. Ranganathan, P. Leech, D. Irwin, and J. Chase, "Ensemble-level power management for dense blade servers," in *IEEE Proceedings of the 33rd International Symposium on Computer Architecture (ISCA'06)*, Boston, MA, May 2006, pp. 66–77.

[9] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs (2nd, extended ed.)*. Springer-Verlag New York, Inc., 1994.

[10] Q. Tang, S. K. S. Gupta, D. Stanzione, and P. Cayton, "Thermal-aware task scheduling to minimize energy usage of blade server based datacenters," in *IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06)*, Oct 2006.

[11] "Flovent CFD simulation software." [Online]. Available: http://www.flomerics.com/

[12] R. K. Sharma, C. E. Bash, and C. D. Patel, "Dimensionless parameters for evaluation of thermal design and performance of large scale data centers," in *Proceedings of the American Institute of Aeronautics and Astronautics (AIAA)*, 2002, p. 3091.

[13] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusam, "Compact thermal modeling for temperature-aware design," in *DAC '04: Proceedings of the 41st annual conference on Design automation*. New York, NY, USA: ACM Press, 2004, pp. 878–883.

[14] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "Hotspot: A compact thermal modeling methodology for early-stage vlsi design." *IEEE Trans. VLSI Syst.*, vol. 14, no. 5, pp. 501–513, 2006.

[15] K. Chen, D. M. Auslander, C. E. Bash, and C. D. Patel, "Local temperature control in data center cooling," Hewlett Packard Laboratories, Tech. Rep. HPL-2006-42, March 2006.

[16] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," *SIGARCH Comput. Archit. News*, vol. 34, no. 2, pp. 78–88, 2006.

[17] R. Ge, X. Feng, and K. W. Cameron, "High-performance, power-aware distributed computing for scientific applications," *IEEE Computer*, pp. 40–47, November 2005.

[18] R. Springer, D. K. Lowenthal, B. Rountree, and V. W. Freeh, "Minimizing execution time in mpi programs on an energy-constrained, power-scalable cluster," in *PPoPP '06: Proceedings of the eleventh ACM SIGPLAN symposium on Principles and practice of parallel programming*. New York, NY, USA: ACM Press, 2006, pp. 230–238.

[19] P. Ranganathan, P. Leech, D. Irwin, and J. Chase, "Ensemble-level power management for dense blade servers," in *ISCA '06: Proceedings of the 33rd annual international symposium on Computer Architecture*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 66–77.

[20] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing server energy and operational costs in hosting centers," in *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM Press, 2005, pp. 303–314.

[21] T. Heath, B. Diniz, E. V. Carrera, W. M. Jr., and R. Bianchini, "Energy conservation in heterogeneous server clusters," in *PPoPP '05: Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming*. New York, NY, USA: ACM Press, 2005, pp. 186–195.

[22] C. D. Patel, R. Sharma, C. E. Bash, and A. Beitelmal, "Thermal considerations in cooling large scale high compute density data centers," in *Proceedings of the Eight Inter-Society Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, San Diego, CA, June 2002, pp. 767–776.

[23] M. H. Beitelmal and C. D. Patel, "Thermo-fluids provisioning of a high performance high density data center," Hewlett Packard Laboratories, Tech. Rep. HPL-2004-146, September 2004. [Online]. Available: http://www.hpl.hp.com/techreports/2004/HPL-2004-146.html

[24] J. Moore, J. Chase, K. Farkas, and P. Ranganathan, "Data center workload monitoring, analysis, and emulation," in *Eighth Workshop on Computer Architecture Evaluation using Commercial Workloads*, February 2005.

[25] J. Moore, J. Chase, and P. Ranganathan, "Weatherman: Automated, online, and predictive thermal mapping and management for data centers," in *3rd IEEE Int'l Conf. Autonomic Computing*, June 2006.