Thermal Aware Task Scheduling With Artificial Neural Network

Xi He
Service Oriented Cyberinfrastruture Lab, Rochester Institute of Technology
Bldg 74, Lomb Memorial Drive, Rochester, NY 14623-5608
Email: xi.he@mail.rit.edu

***Abstract*—Large energy consumption in data centers has become a challenging problem with the emergence of cloud computing and high performance data centers. Efficiently reducing energy cost is one of key issues involved with both optimizing computing resources and maximizing business outcome. In this paper, we present a thermal aware task scheduling to address the energy problem in data centers. The basic idea of our approach is to balance the temperature distribution in data centers, thus implicitly minimizing cooling energy cost in data centers. As an important component of our scheduling system, a thermal model is also developed and implemented using artificial neural network to predict the effect of workload distribution and cooling configuration on thermal distribution in data centers.**

I. Introduction

In recent years, many large scale data centers have been deployed with high density computing clusters and server farms to support high performance scientific applications. However, besides the scientific challenges, many operational issues in data centers need to be addressed. One of those issues is the large energy consumption in data centers. According to U.S. Environmental Protection Agency (EPA), 61 billion kilowatt-hours of power was consumed in data center in 2006, that is 1.5 percent of all US electricity consumption costing around $4.5 billion [1]. In fact, the energy consumption in data centers doubled between 2000 and 2006 and EPA estimates that the energy usage will double again by 2011.

section 4.1 because in 2-dimensional Gaussian classification, we consider

A large scale data center's annual energy cost can be several millions of US dollars. Power and cooling cost is the dominant cost in data centers [2]. In fact, it is reported that cooling costs can be up to 50% of the total energy cost [3]. It is also noted that the life of a computer system is directly related to its operating temperature. Based on Arrhenius time-to-fail model [4], every 10°C increase of temperature leads to a doubling of the system failure rate. Hence, it is recommended that computer components be kept as cool as possible for maximum reliability, longevity, and return on investment [5].

A recent research work [6] indicates that computing nodes' temperature distribution will affect the energy cost of cooling system in data centers. It also shows that minimizing the maximal temperature of all the nodes will minimize the cooling energy cost of a data center. Assume a set of tasks consume a fixed amount of energy in data centers, an appropriate temperature distribution of computing nodes will reduce the maximal temperature of all the nodes and thus significantly conserve cooling energy consumption.

In our study, we present a thermal aware task scheduling to address the energy problem. The contribution of this paper is two-fold: first, we developed a thermal-aware task scheduling for data centers, which will save an amount of cooling energy cost. Second, we develop and implement a thermal model with artificial neural network.

The remainder of this paper is organized as follows: first we introduce three ways to approach the energy problem in data centers in Section 2. Then we discuss the motivation for thermal aware scheduling in Section 3. In Section 4, we present compute nodes, thermal and job model in data centers, and define our problem. The detailed explanation of our thermal model and its implementation is presented in Section 5. We present our algorithm and simulation result in Section 6,7. In the last section, we conclude the paper and propose our future work.

II. Literature review

Researchers usually approach the thermal management problem in three different ways. One is from the infrastructure design and planning perspective. In [7], CFD modeling and increased deployment of temperature sensors are involved in the phase of data center design and analysis. The work in [8] evaluates layout of the computing equipment in the data center to minimize air flow inefficiencies. The second approach aims to improve the computation power efficiency. In our lab's recent publication [9], we focuses on scheduling virtual machines in a compute cluster to reduce power consumption via the technique of Dynamic Voltage Frequency Scaling (DVFS). The third approach, which is applied in this paper, is targeted on improving the cooling power efficiency.

The cooling system in data centers extracts the heat produced by servers and maintains server's inlet air temperature below the redline temperature. Typically the efficiency of the cooling system depends on external environmental control which is affected by irregular air flows and nonuniform workload. Therefore, more efficient cooling system can be achieved by appropriate workload placement. In [10], based on the simulation result that there exists temperature imbalance for a row of racks, the researchers proposes to schedule the workload according to the extract temperature of the racks in a row. In [6], [11], the researchers study the recirculation problem in data centers and propose a task scheduling algorithm to

minimize heat recirculation, thus leading to minimal cooling energy cost. The shortcoming of these two approaches is that instead of compute nodes, they study compute racks which contain a certain number of compute nodes with different temperature. As a result, these solutions are less accurate than others. The more elaborate thermal aware tasks are with computational fluid dynamic (CFD) models [12]. However, some research declares that CFD based model is too complex and is not suitable for online scheduling. HP lab presents a neural network based thermal predictive model [12]. Unfortunately, no detailed information is exposed to academic community due to industry restriction. Also the thermal aware task scheduling proposed in this paper is time-consuming and impractical.

III. Motivation for Thermal Aware Task Scheduling

Temperature is considered as an important physical metrics in data centers [13]. Efficient thermal management not only decrease the cooling costs in data centers, but also increase hardware reliability. In this section, We introduce a typical data center architecture, its nonuniform temperature distribution and the correlation between compute nodes' temperature and their workload. Then we present the motivation for thermal aware task scheduling.

A. Data Center Organization

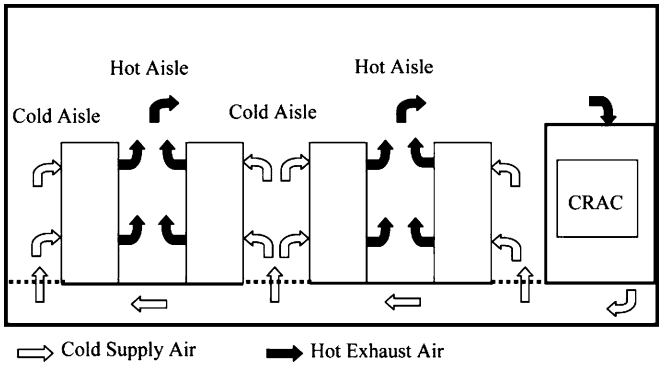

Fig. 1. The typical architecture of data centers

Figure 1 shows how a data center is organized. The racks are laid out in rows on a raised floor over a shared plenum, and arranged back-to-back so that cool aisles and hot aisles are formed to minimize air mixing and increase cooling efficiency. Computer room air conditioning(CRAC) units along the walls take in the re-circulated exhaust hot air, cool the air over a refrigerated or chilled water cooling coil to approximately 10-17° and direct the cooled air into the shared plenum.

B. Thermal Load Distribution

Typically, every data center has its thermal profile inherent to its layout and capability of cooling infrastructure. Due to complex air flow and workload, there exists thermal imbalance in data centers. For example, some locations in the data centers has the higher temperature than others, which we call hot spots. Figure 2 is based on a CFD based study in HP Labs [10] and it shows the contour plot of temperature at a height of 1.85m above the floor in a 11.7m × 8.5m × 3.1m data center. As can be seen from this figure, temperature distribution is not uniform, even the workload distribution is assumed to be uniform in the study. There are several "hot spots" indicated by the regions in red and several "cold spots" indicated by the regions in darker green and blue. The uneven thermal load distribution often leads to an excessive, inefficient cooling cost as it is difficult for current cooling solutions to identify and eliminate "hot spots".

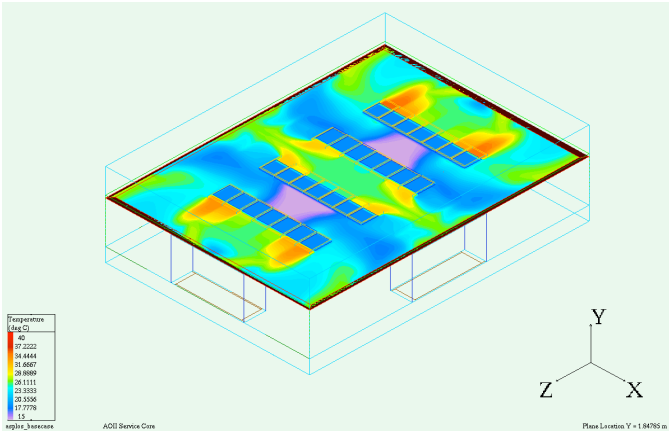

Fig. 2. Temperature contour plot for a programmable data center [10]

C. Task-temperature Profiling

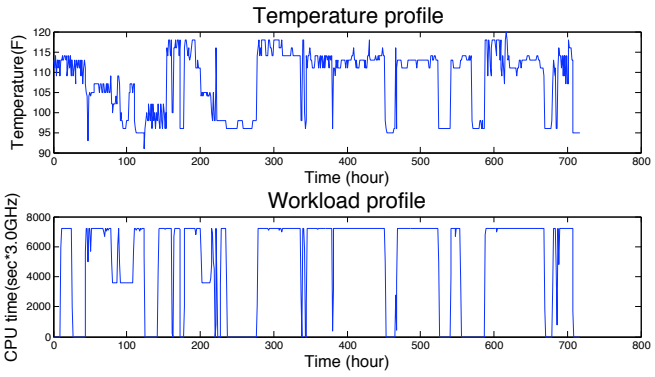

Fig. 3. Task-temperature profiles in buffalo data center

Given certain compute processors and steady ambient temperature, a task-temperature profile shows the temperature increase along with the task execution. Figure 3 shows an overall task-temperature profiles in the Center for Computational Research at State University of New York at Buffalo [14]. The X-axis is the time, and the Y-axis gives two values: the upper line is the task execution time (CPU time), the lower line shows computing node temperature. Figure 3 indicates the task and temperature correlation: as compute loads in term of task CPU time augment, computing node temperatures increase incidentally.

D. Motivation

Based on the above discussion, it is clear that irregular air flow and workload forms nonuniform thermal distribution in data centers, which leads to cooling inefficiency. At the same time, our data analysis shows that the compute nodes' temperature and their workload are strongly correlated. Base on the assumption that uniform thermal distribution can lower cooling cost, it is reasonable that we can conserve cooling energy by the way of scheduling tasks to balance thermal distribution in data centers.

IV. System Model

Before selecting an appropriate technique to implement our idea, we must formalize our problem statement. This section presents formal models of compute resource, job and thermal prediction, and a thermal aware scheduling algorithm, which allocates incoming jobs on compute resources in a data center with the goal of minimizing maximum temperature in the data center.

A. Compute Resource Model

$$Node = \{node_i | 1 \leq i \leq N\} \tag{1}$$

$Node$ represents a set of N compute nodes. $Node_i$ is a compute node which is described as follows:

$$node_i = (< x, y, z >, temp(t), w(t), t^a) \tag{2}$$

where,
$< x, y, z >$ is $node_i$'s location in a 3-dimensional space. t^a is the time when $node_i$ is available for job execution. $temp(t)$ is $node_i$'s temperature-time function.

$$temp(t) = \begin{cases} actual\ temperature & \text{if } t < t^{now} \\ predicted\ temperature & \text{if } t >= t^{now} \end{cases}$$

Note that $temp$ function describes not only $node_i$'s history temperature, but also future temperature that is predicted according to the workload that $node_i$ is going to take. $w(t)$ is a function that represents the workload on $node_i$ over time.

B. Thermal Model

$$T = P(W, L) \tag{3}$$

T represents the thermal topology in a data center. It has non-linear relation P with two factors: W, the workload distribution in the data center and L, the physical topology of the data center.

Since P describe the relationship between temperature and workload, given nodes' workload and their current temperature, we can use P to predict nodes' next moment's temperature.

$$node_i.temp(t+1) = P(node_i.temp(t), node_i.w(t+1)) \tag{4}$$

C. Job model

$$Job = \{job_j | 1 \leq j \leq J\} \tag{5}$$

J is the total number of incoming jobs. job_j is an incoming job, which is described as follows:

$$job_j = (p, t^{arrive}, t^{start}, t^{req}) \tag{6}$$

where,
p is the required compute node number of job_j,
t^{arrive} is the arrival time of job_j,
t^{start} is the starting time of job_j,
t^{req} is the required execution time of job_j.

D. Research Issue Definition

Based on the above discussion, a job schedule is a map from a job job_j to certain compute node $node_i$ with starting time $job_j.start$:

$$schedule_j : job_j \rightarrow (node_i, job_j.t^{start}) \tag{7}$$

A workload schedule $Schedule$ is a set of job schedules $\{schedule_j | job_j \in Job\}$ for all jobs in the workload:

$$Schedule = \{schedule_j | job_j \in Job\} \tag{8}$$

We define the maximum temperature of the compute nodes as $TEMP_{max}$; The problem definition is as follows: given a workload set Job and a set of compute node $Node$, find an optimal workload schedule, $Schedule$, which minimizes $TEMP_{max}$;

V. Thermal prediction using artificial neural network

As stated in subsection IV-B, the thermal model is used to describe the relationship between nodes' temperature and their workload as well as a tool to predict nodes' temperature. Many technologies can be used to implement thermal model, such as a genetic algorithm, support vector machine and artificial neural network.

In recent years, artificial neural networks (ANNs) has been widely applied to a number of prediction problems in different fields, such as "forex prediction" in financial markets or "temperature forecasting" in weather prediction. Thermal topology in data centers changes within minutes and it has the non-linear relationship with its previous thermal topology, workload distribution and effectiveness of cooling system. The temperature in a single compute node not only depends on its previous temperature and its workload, but also its neighborhood compute nodes and its spatial location. Considering ANNs' ability to find the correlation of various variables, we choose ANNs to predict the thermal topology in data centers.

The ANNs approach used in this paper is Back Propagation Neural Network in which learning is performed through the adjustment of errors that are regularly propagated back all the way to the input layer. As shown in Figure 4, the neural network contains 3 types of layers: input layer, hidden layer

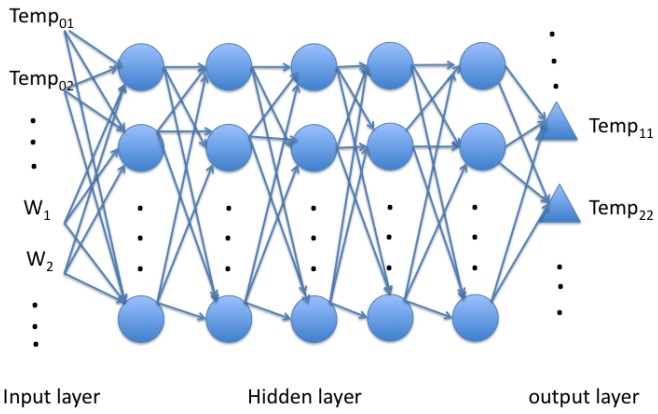

Fig. 4. 5-layer back propagation neural network

and output layer. Input data is composed of two types of data: nodes' previous moment's temperature and their current workload. Nodes' current moment's temperature is predicted in the output layer. Between the input layer and the output layer, there are five hidden layers, and each layer contains a set of elements known as neurons. Each neuron accepts inputs from the previous layer, applies a weighting factor to each input and uses the sum of the weighted inputs as the input of its activation function, which is $tansig$ in our ANNs. Then the output of the neuron's activation function is passed as input to the next layer.

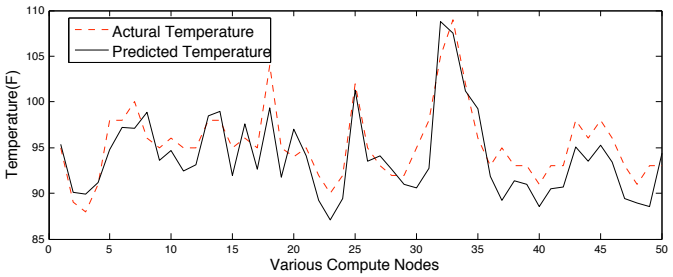

Fig. 5. ANNs simulation result

We select Matlab Neural Network Toolbox to implement the back propagation neural network. We define five hidden layers in our neural network, and each hidden layer contains four hundred neurons and used the $tansig$ function as its activation function. The target Mean Square Error (MSE) is 0.025. The data for training and testing ANNs is from a real data center environment based on the Center for Computational Research (CRR) of State University of New York at Buffalo. We collected nodes' temperature data from the on-board sensors inside the compute nodes, and incoming jobs' information from job logs. We selected 50 compute nodes in our simulation, use the data within 100 hours to train the neural network and then use the well-trained neural network to predict nodes' temperature at the next moment. Figure 5 shows a plot of predicted temperature distribution versus the actual distribution. Over 40% of prediction difference are within $1°C$ and 70% are within $2°C$.

VI. Algorithm

This section discusses our Thermal Aware Scheduling Algorithm (TASA). The key idea of TASA is to schedule "hot" jobs on "cool" compute nodes and tries to minimize maximum temperatures of compute nodes.

Algorithm 1 Thermal Aware Scheduling Algorithm (TASA)

01 $t = 0$
02 FOR $i = 1$ TO N DO
03 $node_i.t^a$=0;
04 ENDFOR
05 Initiate $List_a$
06 FOR $node_i \in Node$ DO
07 Insert $node_i$ into $List_a$, and keep $List_a$ the increased order of $node_i.temp(t^a)$
08 ENDFOR

09 Update $node_i.temp(t)$ with temperature measurement
10 Sort Job in the order of decreased $job_j.t^{req}$

11 FOR $j = 1$ TO J DO
12 FOR $k = 1$ TO $job_j.p$ DO
13 Set var to the first element in $List_a$ and remove var from $List_a$
14 $var.t^a = var.t^a + job_j.t^{req}$
15 Calculate $var.w(t^a - 1)$ with $job_j.t^{req}$
16 Predict $var.temp(t^a - 1)$ with P
17 IF $var.temp(t^{a-1} >= t_{redline})$ THEN
18 WHILE $(var.temp(t^a - 1) >= t_{redline})$ DO
19 Schedule job_{idle} on var
20 $var.t^a = var.t^a + 1$;
21 Calculate $var.w(t^a - 1)$
22 Predict $var.temp(t^a - 1)$ with $var.f()$
23 END WHILE
24 ENDIF
25 Insert var into $List_a$, and keep $List_a$ the increased order of $node_i.temp(t^a)$
26 ENDFOR
27 Schedule job_j on $\{node_{j1}, node_{j2}, \ldots, node_{jp}\}$
28 ENDFOR

29 $t = t + T^{interval}$
30 Accept incoming jobs
31 go to 09

Algorithm 1 presents a Thermal Aware Scheduling Algorithm (TASA). Lines 1– 8 initialize variables. Line 1 sets the initial time stamp to 0. Lines 2 – 4 set compute nodes available time to 0, which means all nodes are available from the beginning. Lines 5 – 8 initialize a list $List_a$ and add all the available compute nodes into the list in the increased order of nodes' temperature.

Lines 9 – 31 schedule jobs periodically with an interval of $T^{interval}$. Line 9 updates each node's $temp(t)$ with temperature measurement. Line 10 sorts the incoming jobs by their execute time. Actually we think a job's execute time is an important factor on determining how hot or how cool a job is. By sorting the incoming jobs by their execute time, we also sort the jobs from "hottest" to "coolest".

Lines 11 – 28 allocate jobs to all compute nodes. Line 11 gets a job from sorted job list, which is the "hottest" job and line 12 allocates the job with a number of required nodes, which are the "coolest". Line 13 – 16 update those allocated nodes' information, such as the next available time, and the predicted temperature at the next available time using thermal model P. Lines 17 – 24 deal with the situation that the node's predicted temperature at the next available time is above "redline", which causes the compute node down. In that case, our algorithm will schedule "empty task" on the node and do not allocate any real task on it until its temperature cools down. Line 25 predicts the temperature of next available time for these allocated nodes. Then these nodes are inserted into $List_a$, which keeps the increased node temperature at next available time.

Algorithm 1 waits a for period of $T^{interval}$ and accepts incoming jobs. It then proceeds to the next scheduling round.

VII. Simulation and Performance Evaluation

A. Simulation Environment

We simulate a real data center environment based on the Center for Computational Research (CCR) of State University of New York at Buffalo. All jobs submitted to CCR are logged during a 30-day period, from 20 Feb. 2009 to 22 Mar. 2009. CCR's resources and job logs are used as input for our simulation of the Thermal Aware Scheduling Algorithm (TASA).

CCR's computing facilities include a Dell x86 64 Linux Cluster consisting of 1056 Dell PowerEdge SC1425 nodes, each of which has two Irwindale processors (2MB of L2 cache, either 3.0GHz or 3.2GHz) and varying amounts of main memory. The peak performance of this cluster is over 13TFlop/s.

The CCR cluster has a single job queue for incoming jobs. All jobs are scheduled with a First Come First Serve (FCFS) policy. There were 22385 jobs submitted to CCR during the period from 20 Feb. 2009 to 22 Mar. 2009. Figure 6, Figure 7 and Figure 8 show the distribution of job execution time, job size (required processor number) and job arrival rate in the log. We can see that 79% jobs are executed on one processor and job execution time ranges from several minutes to several hours.

In the following section, we simulate the Thermal Aware Scheduling Algorithm (TASA) based on the job-temperature profile, job information, thermal maps, and resource information obtained in CCR log files. We evaluate the thermal aware scheduling algorithm by comparing it with the original job execution information logged in the CCR, which is scheduled by FCFS. In the simulation of TASA, we set the maximum temperature threshold to 125 °F.

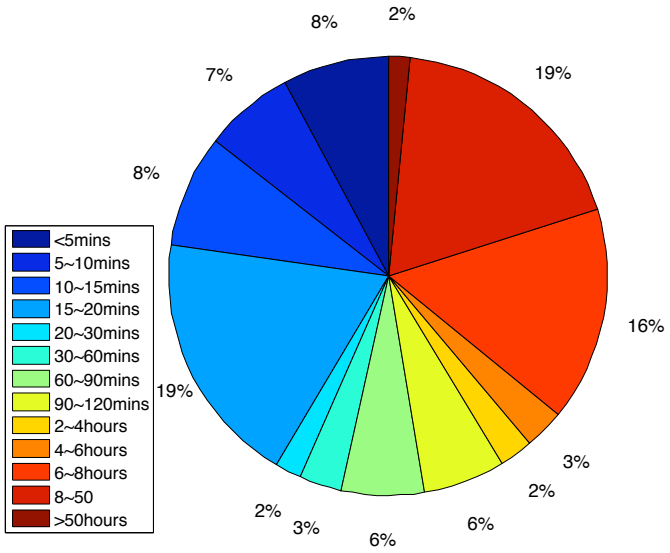

Fig. 6. Job execution time distribution

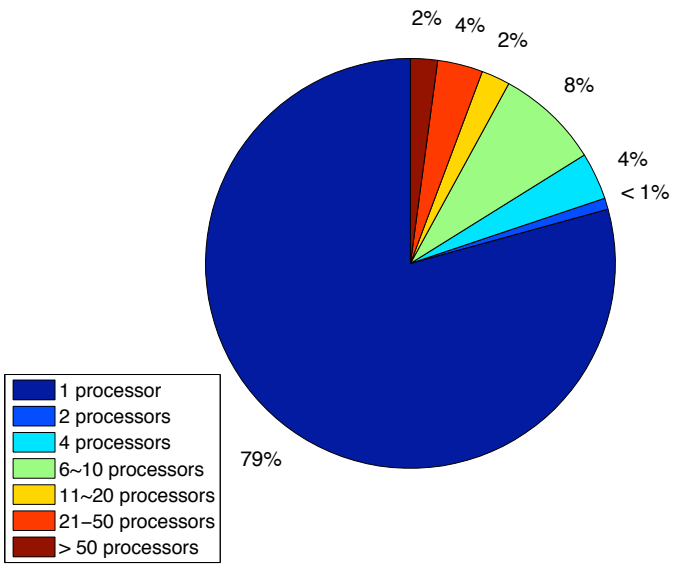

Fig. 7. Job size distribution

B. Experiment Results

1) Data Center Temperature: Firstly we consider the maximum temperature in a data center as it correlates with the cooling system operating level. As shown in Figure 9, the X-axis is the time, and the Y-axis gives two values: the low line is the maximum temperature using TASA; the upper line is the maximum temperature using FCFS. Compared with FCFS, the maximum temperature reduced by TASA is 19 °F and the average temperature reduced by TASA is 12 °F.

2) Job response time: We have reduced power consumption and have increase the system reliability, both by decreasing the data center temperatures. However, we must consider that there may be trade offs by an increased response time.

The response time of a job $job_j.t^{res}$ is defined as job execution time ($job_j.t^{req}$) plus job queueing time ($job_j.t^{start}$ –

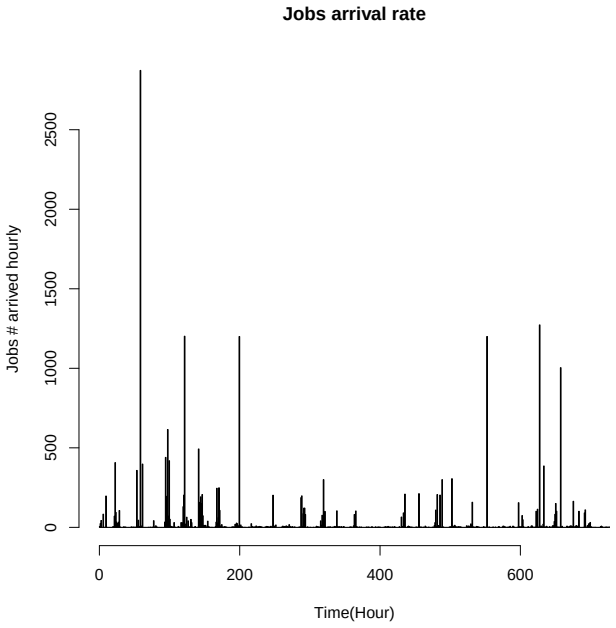

Fig. 8. Job arrive rate distribution

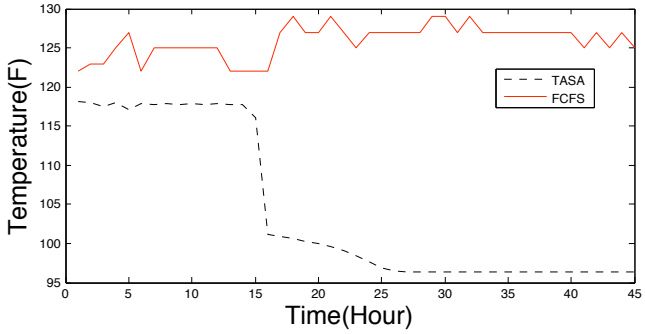

Fig. 9. Comparison of maximum temperature

$job_j.t^{arrive}$), as shown below:

$$job_j.t^{res} = job_j.t^{req} + job_j.t^{start} - job_j.t^{arrive} \quad (9)$$

To evaluate the algorithm from the view point of users, job response time indicates how long it takes for job results to return to the users.

As the thermal aware scheduling algorithm intends to delay scheduling jobs to some over-hot compute nodes, it may increase the job response time. In the simulation, we got the $overhead = 11\%$. Which means that we reduce the 19 °F of temperature in CCR data center by paying cost of increasing 11% job response time.

VIII. Conclusion

To improve energy efficiency and reliability of data center operation, we study the temperature distribution in data centers and present our thermal aware task scheduling. As an important step to practically implement our algorithm, we apply artificial neural network technique to predict the effect of workload distribution and cooling configuration on temperature distribution in data centers. We also conduct a simulation to compare the thermal efficiency between TASA and classic FCFS. Compared with FCFS, TASA decreases the maximum temperature in the data center by 19 °F with the cost of increasing 11% job response time.

In the future work, we are interested to improve our neural network model and pay more attention to the effect of compute nodes' spatial location on temperature distribution. We also plan to compare our neural network based prediction model with CFD based prediction model. In addition, as backfilling algorithm is popular in parallel systems, we would integrate back-filling algorithm into our thermal aware scheduling algorithm to improve system performance.

References

[1] "Report to Congress on Server and Data Center Energy Efficiency ." [Online]. Available: http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf

[2] "The green grids opportunity: decreasing datacenter and other IT energy usage patterns," the green grid, Tech. Rep., Feb. 2007.

[3] R. Sawyer, "Calculating Total Power Requirements for Data Centers," American Power Conversion, Tech. Rep., 2004.

[4] P. W. Hale, "Acceleration and time to fail," *Quality and Reliability Engineering International*, vol. 2, no. 4, pp. 259–262, 1986.

[5] "Operating Temperature vs System Reliability," Website, 2009. [Online]. Available: http://www.pcpower.com/technology/optemps/

[6] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Thermal-aware task scheduling for data centers through minimizing heat recirculation," in *CLUSTER*, 2007, pp. 129–138.

[7] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," in *DAC*, 2004, pp. 878–883.

[8] R. Sullivan, "Alternating cold and hot aisles provides more reliable cooling for server farms," *Uptime Institute*, 2000.

[9] G. von Laszewski, L. Wang, A. J. Younge, and X. He, "Power-aware scheduling of virtual machines in dvfs-enabled clusters," in *IEEE Cluster 2009*, IEEE. New Orleans, Louisiana: IEEE, 08/2009 2009.

[10] R. K. Sharma, C. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase, "Balance of power: Dynamic thermal management for internet data centers," *IEEE Internet Computing*, vol. 9, no. 1, pp. 42–49, 2005.

[11] J. D. Moore, J. S. Chase, P. Ranganathan, and R. K. Sharma, "Making scheduling "cool": Temperature-aware workload placement in data centers," in *USENIX Annual Technical Conference, General Track*, 2005, pp. 61–75.

[12] J. Choi, Y. Kim, A. Sivasubramaniam, J. Srebric, Q. Wang, and J. Lee, "A CFD-Based Tool for Studying Temperature in Rack-Mounted Servers," *IEEE Transactions on Computers*, vol. 57, no. 8, pp. 1129–1142, 2008.

[13] H. Hamann, M. Schappert, M. Iyengar, T. van Kessel, and A. Claassen, "Methods and techniques for measuring and improving data center best practices," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. ITHERM 2008. 11th Intersociety Conference on*, 2008, pp. 1146–1152.

[14] "the Center of Computational Research." [Online]. Available: http://www.ccr.buffalo.edu/display/WEB/Home