Thermal-Aware Task Scheduling to Minimize Energy Usage of Blade Server Based Datacenters*

Qinghui Tang
Dept. Electrical Eng.
Arizona State University

Sandeep. K. S. Gupta, Daniel Stanzione
Dept. Computer Sc.& Eng.
Arizona State University
sandeep.gupta@asu.edu

Phil Cayton
Intel Corporation
Hillsboro OR 97124

Abstract

Blade severs are being increasingly deployed in modern datacenters due to their high performance/cost ratio and compact size. In this study, we document our work on blade server based datacenter thermal management. Our goal is to minimize the total energy costs (usage) of datacenter operation while providing a reasonable thermal environment for their reliable operation. Due to special characteristics of blade servers, we argue that previously proposed power-oriented schemes are ineffective for blade server-based datacenters and that task-oriented scheduling is a more practicable approach since the contribution to the total energy cost from cooling and computing systems vary according to the utilization rates. CFD simulations are used to evaluate scheduling results of three different task scheduling algorithms: Uniform Outlet Profile (UOP), Minimal Computing Energy (MCE), and Uniform Task (UT), under four different blade-server energy consumption models: DiscreteNonOptimal (DNO), DiscreteOptimal (DO), AnalogNonOptimal (ANO), and AnalogOptimal (AO). Simulation results show that the MCE algorithm, in most cases, results in a minimal total energy cost - a conclusion that differs from the findings of previous research. UOP performs better than UT at low datacenter utilization rates, whereas UT outperforms UOP at high utilization rates.

1 Introduction

Computing clusters and server farms are increasingly housed in datacenters that are limited by power and thermal capacity. For a large scale datacenter, the annual energy cost can run into millions of dollars, and the cooling cost is at least half of the total energy cost [7]. Improperly designed or operated datacenters may either suffer from overheated servers and potential system failures, or from overcooled systems and higher utilities costs. Thus, minimizing the energy cost and improving thermal performance of datacenters is one of the key issues towards optimizing costs for computing resources and maximally utilizing the installed computation capability of the datacenter.

Thermal performance of a datacenter is a critical metric towards reliably operating a datacenter. It is defined by three criteria: heat dissipation capability, thermal distribution, and temperature variation patterns. Further, lowering operations cost, and extending the life span of electronic equipment are key design objectives that can be achieved through improved thermal performance. From a holistic perspective, there are two major steps for improving the thermal performance of a datacenter. The *first* is from the infrastructure design and planning perspective: Datacenter design and analysis have become increasingly sophisticated, involving computational fluid dynamic (CFD) modeling in the design phase and increased deployment of temperature sensors and supplementary cooling systems. Furthermore, deployment of power-aware computing systems, which have both low power cores and the ability to dynamically change their clock speeds, helps to reduce heat generation. The *second*, which is the focus of this work, is to improve and optimize thermal performance during the operation of a datacenter. More specifically, in this paper, we study thermal-aware scheduling to reduce the total energy cost for operating a datacenter.

The centralized nature of datacenters enables IT administrators to manage, configure and maintain hardware and software systems more efficiently. However, in many cases, it is extremely critical to maintain the datacenter with a desired working environment to keep all application systems running uninterrupted. *Autonomic computing* has been proposed for IT systems which integrates multimodal information to achieve desirable properties such as high reliability, self-manageability and maximized system throughput. Thermal manage-

*This work is supported in part by a grant from Intel Corporation.

ment of datacenters is an essential part of making datacenter management *autonomic*. This includes automatically adjusting task scheduling or assigning tasks according to thermal distributions obtained either through online measurements and/or thermal simulation.

Blade servers are increasingly deployed in the new generation of datacenters due to their high performance/cost ratio and compact size. Blade servers are ideal for specific applications such as web hosting and cluster computing. But it also raises new issues due to the extremely high heat dissipation (more than 2000 W/ft^2 [2]) and therefore creates higher demands on cooling systems. Previous work [5] has studied thermal-aware scheduling of datacenters for reducing cooling costs. However, it does not consider the new power consumption characteristics of blade servers, which include relatively large startup power consumption and multiple power states due to multiple processors being integrated on each blade and multiple blades in each enclosure. We refer to each such enclosure (chassis or blade server) as a **computing node**. Hence, each computing node may have several processors.

In addition, the *power-oriented* scheduling of [5] is not practical as power consumption is not a system variable that can be manipulated directly by system administrators. In this work, we propose a *task-oriented* thermal-aware scheduling to reduce *total energy costs*, and not just the *cooling costs*.

For this work we built a thermal model of a small scale, blade server based datacenter by using the typical power consumption characteristics of a Dell PowerEdge blade server. CFD simulations were used to evaluate scheduling results of three different task scheduling algorithms: Uniform Outlet Profile (UOP), Minimal Computing Energy (MCE), and Uniform Task (UT), under four different blade-server energy consumption models: DiscreteNonOptimal (DNO), DiscreteOptimal (DO), AnalogNonOptimal (ANO), and AnalogOptimal (AO). Simulation results show that the MCE algorithm, in most cases, results in a minimal total energy cost - a conclusion that differs from the findings of previous research. UOP performs better than UT at low datacenter utilization rates, whereas UT outperforms UOP at high utilization rates.

2 Problem Statement

The datacenter system is composed of N computing nodes, identified as Nodes 1 to N. These nodes work individually or cooperatively to accomplish assigned tasks. For simplicity, we assumed that all tasks are identical and one task is assigned to one processor. A scheduler dispatches the total set of tasks C to individual computing nodes depending on various scheduling policies. Node i consumes power at the rate P_i while performing the task set C_i (a subset of C). The power consumption rate depends on the hardware characteristics of the node and the task profile (e.g., compute intensive or IO intensive), i.e.,

$$P_i = G_i(C_i), \tag{1}$$

where G_i depends on the hardware specifications of computing nodes.

Servers are cooled by using traditional air-cooled technology. Conceptually, each node i's fan draws cold air over the node i at flow rate f_i and inlet temperature T^i_{in}, and dissipates heated air with average outlet temperature T^i_{out}. According to the law of energy conservation and the fact that almost all 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 approximated as

$$P_i = \rho f_i C_p \left(T^i_{out} - T^i_{in}\right), \tag{2}$$

where C_p is the specific heat of air and ρ is the air density. In short, the power consumption of node i will cause air temperature to rise from T^i_{in} to T^i_{out}.

2.1 Total Energy Cost of Datacenter

The total energy cost, or usage, of datacenters 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 is not considered due to its negligible contribution to the total energy cost. The total computing power consumption P_c is presented as

$$P_c = \sum_{i=1}^{N} P_i. \tag{3}$$

The energy cost of operating a cooling device depends on the heat removed and the **Coefficient Of Performance (COP)** of the cooling device. COP 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 2 indicates that to remove 1000 W heat, the work performed by the cooling device is 500 W. The COP is not constant and normally increases with the supplied air temperature. We use the COP model used in [5], which is obtained from a water-chilled CRAC unit in HP Utility Data Center

$$COP = (0.0068T^2_{\text{sup}} + 0.0008T_{\text{sup}} + 0.458), \tag{4}$$

where T_{sup} is the supply air temperature. The cooling cost can be described as $P_{AC} = \frac{P_c}{COP}$. The total energy consumption for operating a datacenter is defined as:

$$P_{Total} = P_{AC} + P_c \tag{5}$$

$$= (1 + \frac{1}{COP}) \sum_{i=1}^{N} G_i(C_i). \tag{6}$$

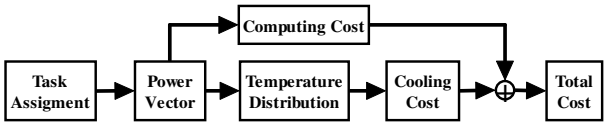

Figure 1. Visualization of thermal-aware scheduling process.

Our goal is to minimize P_{Total}, while satisfying the constraint $C_{Total} = \sum_{i=1}^{N} C_i$:

$$\min \left[(1 + \frac{1}{COP}) \sum_{i=1}^{N} P_i \right] \quad (7)$$

$$\text{subject to : } C_{Total} = \sum_{i=1}^{N} C_i. \quad (8)$$

A task assignment $\vec{\mathbf{C}} = \{C_1, C_2, ...C_N\}$ will result in different power distribution vector $\vec{\mathbf{P}} = \{P_1, P_2, ...P_N\}$, which will lead to inlet/outlet temperature distribution $\overrightarrow{\mathbf{T}_{in}} = \{T_{in}^1, T_{in}^2, ...T_{in}^N\}$ and $\overrightarrow{\mathbf{T}_{out}} = \{T_{out}^1, T_{out}^2, ...T_{out}^N\}$, respectively. We can deliberately reduce cooling cost by rising supplied cold air temperature meanwhile keeping the maximal inlet temperature below the *redline temperature* of devices, which is normally $25°C$ (i.e., $\max \left[\overrightarrow{\mathbf{T}_{in}}\right] < 25°C$). In summary, the problem is how to divide and input task set C, into a task vector $\vec{\mathbf{C}} = \{C_1, C_2, ...C_N\}$ to achieve the minimal total operation energy cost. This process can be visualized as the flow chart shown in Figure 1.

2.2 Energy Model of Blade Servers

Unlike traditional 1U or 2U servers, blade servers normally integrate multiple blades into each chassis in which blades (processors) share the common power supply and cooling fan. Each blade itself may have multiple high-performance processors. Thus, a blade server's power consumption characteristic differs from traditional servers, since the chassis itself consumes a significant amount of energy.

The blade server model used in our datacenter study is Dell PowerEdge 1855. Its 7U (12.25-inch) modular chassis can hold 10 Xeon dual-processor EM64T (Extended Memory 64-bit Technology) blade servers. When performing a typical High Performance Computing (HPC) application, the full utilization of 10 blades (the whole chassis) has a total power consumption (including CPU, disk and IO) of 4638 W. Powering on one chassis enclosure only, without powering on any blade servers, has a *startup power consumption* of 630 W. Assigning a task to an idle processor on a powered-on blade

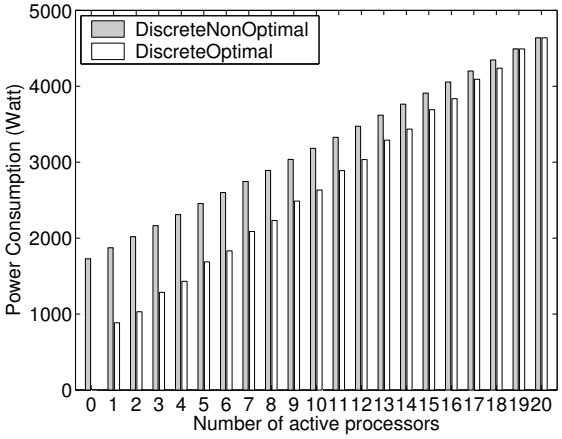

Figure 2. Discrete power states of Dell PowerEdge 1855 blade server. In an ideal optimal case, all idle blades and idle chassis will be shut off. In a non-optimal case, idle blade and idle chassis will be on.

consumes 145.5 W, while assigning a task to a processor on a powered-off blade has power consumption of 145.5 W for the processor plus 109.8 W for the blade module. Therefore, *the power consumption cost of adding a task to one chassis may not be the same as another, since the new assignment may involve waking up an idle chassis or an idle blade.* This characteristic and the aforementioned startup power consumption leads to a different power consumption profile of blade servers compared to traditional servers.

We assume that the chassis can work in two different power consumption modes. The *first mode* is the **optimal mode**, in which any idle blades or idle chassis will be shut-off to avoid unnecessary power wastage. The penalty is the extra start-up time when a new assignment arrives and a higher possibility of component failures. Thus, when the number of processors running at full utilization in a the chassis is C_i, the total power consumption of the chassis is

$$P_i = G_i(C_i) = 630 + 145.5C_i + 109.8 \lceil C_i/2 \rceil . \quad (9)$$

The *second mode* is called the **NonOptimal mode**, in which an idle blade or an idle chassis will not be shut down. When the number of processors running at full utilization is C_i, the total power consumption of the chassis is $630 + C_i * 145.5 + 10 * 109.8$, or

$$P_i = G_i(C_i) = 1728 + 145.5C_i. \quad (10)$$

Figure 2 shows the power consumption of these two modes with an increasing number of processors running

Table 1. Symbols and Definitions.

Symbol	Definition
N	The number of computing nodes
C_p	Specific heat of air – $1005 J/kg/K$
ρ	Density of air – $1.19 kg/m^3$
C_i	Amount of task assigned to server i
P_i	Power consumption of node i
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 – 520 CFM/s
Q_i	Heat dissipation of node i
M_T	Maximal task can be assigned to a node

at full utilization rate. Obviously, when the total utilization of the chassis is low, the Optimal mode can save a significant amount of energy compared to the NonOptimal mode. In addition, the power consumption of chassis will lead to a minimal startup power consumption of 630 W (optimal mode) or 1728 W (nonoptimal mode). This startup power consumption and multiple power states are two main reasons why previous work [4] cannot be applied to blade-server based datacenters.

In our analysis we also assume that tasks can be assigned to blades with **discrete mode** or **analog mode** (utilization rate of 15.5 means the sixteenth processor is running at 50% utilization rate)[1], so we can compare four different *blade server energy models*: **DiscreteNonOptimal (DNO)**, **DiscreteOptimal (DO)**, **AnalogNonOptimal (ANO)** and **AnalogOptimal (AO)**.

3 Thermal Aware Scheduling

In this section, we provide our analysis of three different scheduling algorithms: **Uniform Outlet Profile (UOP)**, **Minimal Computing Energy (MCE)**, and **Uniform Task (UT)**. The granularity of temperature measurement is at the chassis level and the task scheduling granularity is at the processor level. Once we obtain a task assignment result, we map it into the power consumption by using Eq. (1), then we use CFD simulations to evaluate the thermal distribution of the given power vector. The symbols we use in this analysis are listed in Table 1.

[1]In the digital mode, a processor is either busy (100% utilization) or idle (0% utilization). In the analog mode, a processor usage can be any value between 0% to 100%.

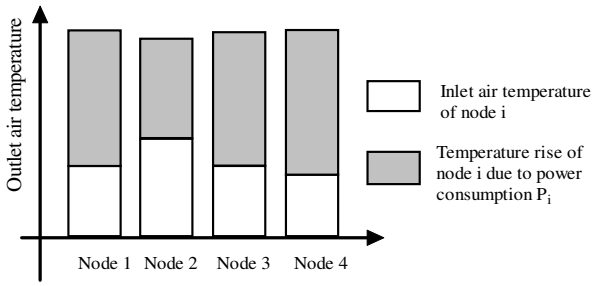

Figure 3. Outlet temperature equals inlet temperature plus the temperature rise due to power consumption P_i. Outlet temperature balancing can be achieved by deliberately assigning tasks.

3.1 Uniform Outlet Profile (UOP)

Based on the inlet temperature of each computing node[2], this algorithm will assign more tasks to nodes with low inlet temperatures, and fewer tasks to nodes with high inlet temperature. The objective is to achieve a uniform outlet temperature distribution. Figure 3 gives a conceptual view of this approach.

To achieve outlet thermal balancing, ideally all nodes' outlet temperature T^i_{out} should have the same value T_c. Considering Eq. (1) and Eq. (2) we have heat transfer of a computing node as

$$G_i(C_i) = \rho f_i C_p \left(T_c - T^i_{in}\right). \tag{11}$$

We sum Eq. (2) over all node i, to get

$$\sum_{i=1}^{N} G_i(C_i) = \sum_{i=1}^{N} \rho f_i C_p \left(T_c - T^i_{in}\right), \tag{12}$$

or

$$T_c = \frac{\sum_{i=1}^{N} G_i(C_i) + \sum_{i=1}^{N} \rho f_i C_p T^i_{in}}{\sum_{i=1}^{N} \rho f_i C_p}, \tag{13}$$

where only C_is ($i = 1$ to N) are unknown and they satisfy the constraint in Eq. (8). For a homogeneous datacenter environment, where all the computing nodes have the same power consumption profile and hardware specification, e.g., $f_i = f$ and $G_i(\cdot) = G_0(\cdot)$, so we have

$$T_c = \frac{G_0(C_i)}{B} + T^i_{in}, \tag{14}$$

where we define constant $B = \rho f C_p$.

[2]A blade-server chassis is modeled as having a single air inlet and a single outlet, temperature is calculated as the average temperature over the entire inlet/outlet.

If we further assume the relationship between P_i and C_i is linear, $P_i = a + bC_i$, as the energy model of a blade server as we discussed in Section 2.2, we get

$$T_c = \frac{Na + bC_{Total} + B\sum_{i=1}^{N} T_{in}^i}{NB}. \quad (15)$$

Further, we can obtain the aggregate power consumption of all the node by Eq. (11)

$$P_i = B\left(T_c - T_{in}^i\right), \quad (16)$$

and consequently we can obtain the tasks assigned to all nodes by the equation: $C_i = (P_i - a)/b$, and the total computing energy cost would then be

$$P_c = NB\left(T_c - T_{in}^i\right). \quad (17)$$

In the discrete energy model, we need to approximate the assigned power to the nearest discrete power level. In addition, in some cases the calculated value of P_i is less than the minimal startup power consumption or greater than the maximal possible power consumption (e.g., 4638 W). In these cases, we adjust P_i to be either minimal or maximal value as appropriate. Consequently the outlet temperature for all the nodes will not be maintained at T_c. So we use the variance of the outlet temperature T_{out}^i as the measurement to decide whether we achieved outlet thermal balancing

$$\min\left[Var\left(\overrightarrow{\mathbf{T}_{out}}\right)\right], \quad (18)$$

where T_{out}^i can be calculated as $T_{out}^i = P_i/B + T_{in}^i$.

Discussion: Although this approach seems similar to the OnePassAnalog algorithm presented in [5], it is fundamentally very different. The OnePassAnalog assigns power consumption based on a reference power budget as

$$P_i = \frac{T_{ref}^{out}}{T_i^{out}} P_{ref}, \quad (19)$$

where T_{ref}^{out} is the reference (average) temperature and P_{ref} is the reference (average) power consumption. OnePassAnalog gives the server low power consumption if it has a high outlet temperature. Although it seems reasonable, this will not work for blade servers because 1) OnePassAnalog tends to assign tasks to (and consequently activate) all the computing nodes, and the assigned power P_i may be even smaller than the minimal startup power consumption of the chassis (630 W or 1728 W); 2) the total power consumption

$$\sum_{i=1}^{N} P_i = \sum_{i=1}^{N} \frac{T_{ref}^{out}}{T_i^{out}} P_{ref}, \quad (20)$$

may not satisfy the constraint $C_{Total} = \sum_{i=1}^{N} C_i$. The root reason is that OnePassAnalog is a power-oriented scheduler, not a task-oriented scheduler as UOP. *In practice, the total tasks, not the total power consumption needs to be divided to groups of computing nodes.*

3.2 Minimal Computing Energy

Minimal Computing Energy (MCE) minimizes the number of powered-on chassis and processors to concentrate computing energy costs on those active servers and processors and turn-off all other idle processors or blades. Consequently, the resulting outlet temperatures of all computing nodes will not necessarily be equal. To reduce the thermal risk, the computing nodes with the lowest inlet temperature will be assigned tasks first. This is technically similar to the *CoolestInlet* approach mentioned in [5].

Let M_T be the maximal amount of tasks that can be assigned to a chassis, then the number of chassis required is $k = \left\lceil \frac{C_{Total}}{M_T} \right\rceil$, where $\lceil x \rceil$ is the function of rounding x to nearest integer greater than or equal to x. The number of tasks assigned to first $k-1$ nodes will be $(k-1)M_T$. The k^{th} node will be assigned the amount of task as $C_{Total} - (k-1)M_T$. Without the loss of generality, we can assume that node 1 to k are the first k nodes with lowest inlet temperature. The total computing energy cost would be

$$P_c = G_i(C_{Total} - (k-1)M_T) + (k-1)\,G_i(M_T). \quad (21)$$

3.3 Uniform Task (UT)

With this scheme, all nodes are assigned the same amount of tasks. Consequently the power consumption for each node i will be

$$P_i = G_i(\frac{C_{Total}}{N}). \quad (22)$$

Similarly, for homogeneous nodes and linear relation between P_i and C_i, we have

$$P_i = G_i(\frac{C_{Total}}{N}) = a + b\frac{C_{Total}}{N} \quad (23)$$

and

$$P_c = \sum_{i=1}^{N} P_i = Na + bC_{Total}. \quad (24)$$

3.4 Difference with previous work

Researchers at HP Labs and Duke University have published work [6] [3] on smart cooling techniques for datacenters. They have developed online measurement and control techniques to improve energy-efficiency of datacenters. They defined Supply Heat Index (SHI) and Return Heat Index (RHI) to characterize the energy efficiency of datacenter cooling system. From a mechanical or civil engineering perspective, they discussed how different datacenter layouts and configurations will lead to different thermal distributions and hotspots. Based on a CFD simulation model, they also determined which area

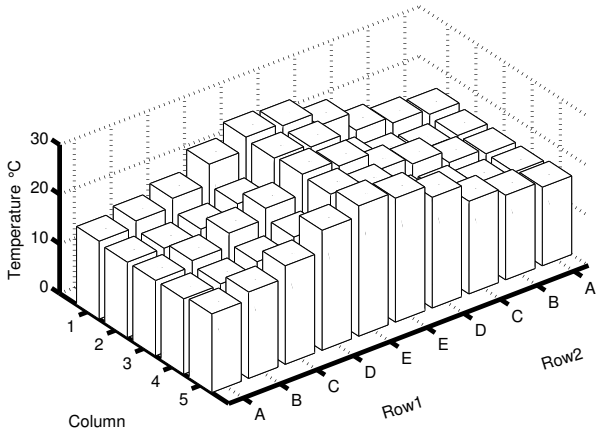

Figure 4. Inlet temperature distribution: chassis located at the lower part of the rack obtain plenty of cold air from floor vents and have a low inlet temperature.

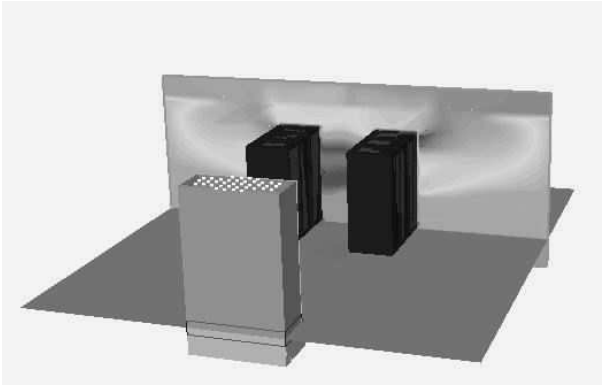

Figure 5. Datacenter model used in our study: two rows of standard 42U racks, one typical underfloor supplied computer room air conditioner. Exhausted warm heat returns from the ceiling vent tiles.

inside a datacenter is overheated. Appropriate remedial actions are then performed accordingly. They also proposed dynamic relocation of workload to achieve thermal balancing in [5] and [4].

The balance of workload, considered in the aforementioned research approaches, is based on the power consumption and not the utilization rate or task load on the computing nodes. As we discussed in Section 3.1, a datacenter administrator does not receive the straight workload in terms of watts but computes tasks in terms of how many resources/processors are required. So a **task-oriented, instead of power-oriented scheduling** approach, is more appropriate to be applied in datacenter.

In addition, due to different power consumption characteristics of blade servers, such as startup power consumption and multiple power status, transferring a certain amount of load from node A to node B will not have the same amount of power consumption drop/increase in node A/B. This is because the power consumption change not only depends on the amount of load changed but also depends on the node's previous power status, e.g., an assignment may involve waking up an idle chassis or an idle blade.

4 Simulation

We used Flovent [1], a CFD simulation software to conduct thermal simulation to obtain the thermal distribution of scheduling results. The datacenter we simulated is a small scale datacenter with physical dimensions $9.6m \times 8.4m \times 3.6m$. It has two rows of industry standard $42U$ racks arranged in a typical cold aisle and hot aisle layout. The $15°C$ cold air 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 the exhausted hot air return to air conditioner through ceiling vent tiles. There are 20 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 is 2000 processors.

For simplicity, we assumed that the total amount of computing task is the number of processors required. A task equal to 20 means the task requires to be performed with 20 processors, or 10 dual-processor blades, or a whole PowerEdge 1855 chassis. A 10% utilization rate of a data center utilization rate means 200 processors are running at full utilization rate. Figure 5 shows the 3D model of the datacenter.

Figure 4 shows the inlet temperature distribution when all the servers are idle. Obviously, the chassis located at the lower part of the rack (A and B) obtain plenty of cold air from the floor vent and have a lower inlet temperature, where chassis located at the upper part (E) of the rack experience highest inlet temperatures due to the insufficient supply of cold air.

4.1 Simulation Results

We plotted the total energy cost against various data center utilization rate as shown in Figure 6 and Figure 7. The numerical results of analog-based models are not shown in figures due to space limitation. From these results, we can observe the computing energy cost increases linearly with the increase of utilization rate, whereas cooling cost increases exponentially due to the

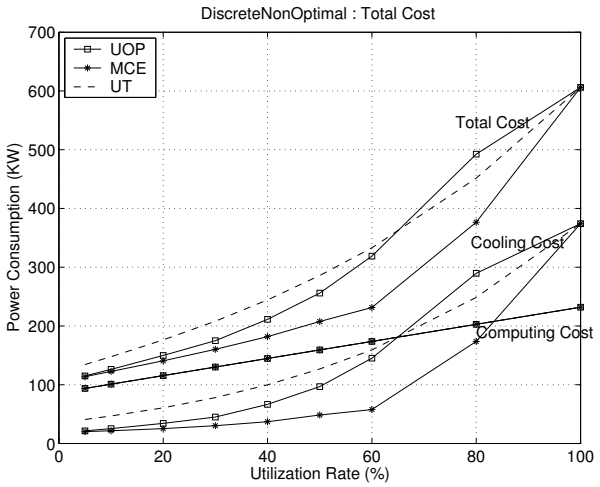

Figure 6. DiscreteNonOptimal

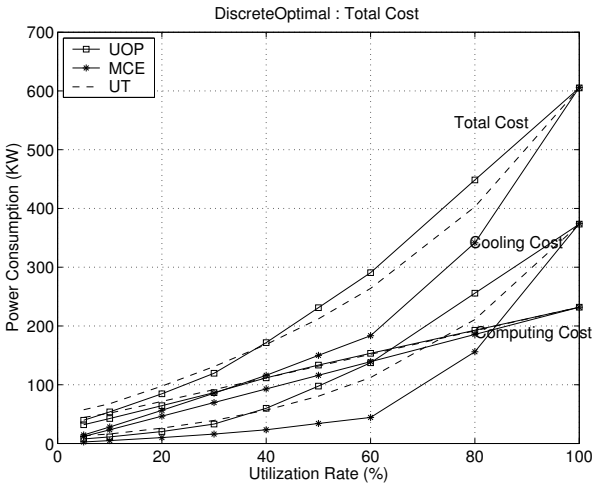

Figure 7. DiscreteOptimal

nonlinearity of COP as shown in Eq. (4). Normally, when the utilization is less than 60%, the dominant part of total energy cost is contributed by compute energy. Once the utilization rate exceeds 60%, the cooling cost replaces the computing cost as the most significant part. The differences among computing costs of the three scheduling algorithms are not as prominent as the difference of cooling cost.

Within a specific energy model, the performance of the three different algorithms vary. For DiscreteNonOptimal, MCE always has the minimal total energy cost, whereas UOP is better than UT at high utilization rates, UT outperforms UOP at low utilization rates. In DiscreteOptimal, such advantage can be seen even when the utilization rate is only about 40%.

The following tables show numerical values of the total cost, the computing cost and the cooling cost when the utilization rate is 60% for different energy models and scheduling algorithms. Energy efficiency for optimal models is better than nonoptimal models. For a given energy model, the computing cost is almost the same for all algorithms, whereas cooling cost varies significantly according to the algorithms.

Total Cost

(KW)	DNO	DO	ANO	AO
UOP	319	291	365	294
UT	332	264	333	264
MCE	231	183	231	183

Computing Cost

(KW)	DNO	DO	ANO	AO
UOP	174	153	173	154
UT	174	151	173	152
MCE	174	139	173	138

Cooling Cost

(KW)	DNO	DO	ANO	AO
UOP	145	145	145	145
UT	159	159	159	159
MCE	58	58	58	58

Even though MCE is the most energy efficient algorithm, it has some practical limitations. This is because under the MCE algorithm, the chassis at the lower part of the rack will be used excessively and will experience higher hardware failure rate due to unremitting long time operation. Our future work will consider hardware reliability models and hardware cost models to address this issue, we will balance the trade-off between energy cost, hardware failure cost, and resulting labor cost of replacing or repairing hardware.

Figure 8 shows three different outlet temperature distributions of DiscreteNonOptimal when the utilization rate is 50%. UOP has a relatively uniform outlet temperature distribution; MCE tries to assign tasks to the coolest inlet, so the chassis located at the lower part of the rack will have higher outlet temperature; for UT, all chassis experience the same temperature rise hence the outlet temperature has similar distribution as the inlet temperature as shown in Figure 4.

Figure 9 shows the standard variation in the outlet temperature against different utilization rates for the DiscreteNonOptimal model. UOP always has the minimal temperature variation due to the balanced outlet temperature. The minimal value is achieved for the mid utilization rates. When the total utilization rate is too low or too high, it is relatively hard to achieve better balancing. With a low utilization rate, we do not have enough tasks and consequently enough power consumption to narrow the temperature gap between different nodes. With a high utilization rate, all the nodes with lower outlet temperatures are running at full utilization rate and cannot accept more tasks. Thus it is inevitable

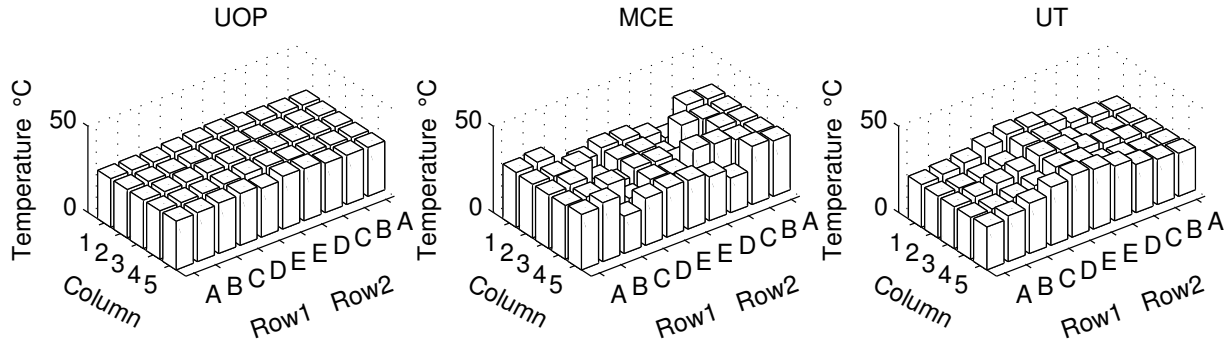

Figure 8. Comparison of outlet temperature distribution: UOP has an almost uniform distribution. For MCE, the lower part chassis experience higher outlet temperatures.

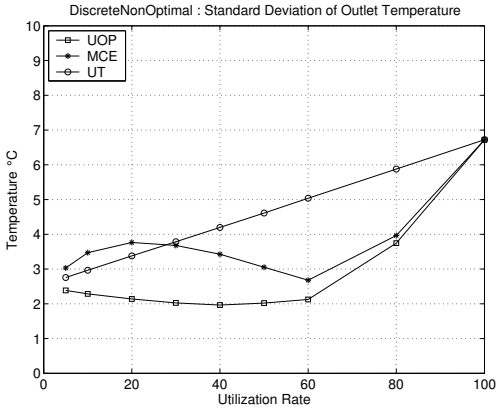

Figure 9. Standard variation of outlet temperature with DNO energy mode: UOP has a minimal temperature variance due to balanced outlet temperatures.

that chassis with high inlet temperatures will receive tasks and generate high outlet temperatures.

5 Conclusions and Future Work

In this work we studied thermal-aware task scheduling of blade server based datacenters to reduce overall energy costs. After considering the power consumption characteristics of blade servers and a practical and task-oriented, instead of power-oriented, scheduling, we showed that MCE is the most energy efficient scheme if the hardware reliability is not considered. UOP performs better than UT at low datacenter utilization rates, where UT outperforms UOP at high utilization rates. Although the case study is based on a homogeneous datacenter with linear energy model of blade servers, the analysis and problem formalization in our work can also be applied to nonlinear energy models and heterogeneous datacenters.

Our analysis and simulation in this work did not consider the fact that the inlet temperature will change with change in power consumption assignment due to complicated air circulation. The recirculation of warm air is also considered to further reduce cooling cost in [5]. In further studies we will conduct a more comprehensive analysis considering the impact of the recirculation of warm air.

References

[1] Flovent CFD simulation software.

[2] ASHRAE. Thermal guidelines for data processing environments. Guideline, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 2004.

[3] M. H. Beitelmal and C. D. Patel. Thermo-fluids provisioning of a high performance high density data center. Technical Report HPL-2004-146, Hewlett Packard Laboratories, September 2004.

[4] 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.

[5] 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.

[6] 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)*, page 767C776, San Diego, CA, June 2002.

[7] R. Sawyer. Calculating total power requirements for data centers. White Paper, 2004.