Energy Efficiency of Thermal-Aware Job Scheduling Algorithms under Various Cooling Models

Georgios Varsamopoulos, Ayan Banerjee, and Sandeep K.S. Gupta

The Impact Laboratory,
Arizona State University
Tempe, AZ 85287, USA
http://impact.asu.edu/

Abstract. One proposed technique to reduce energy consumption of data centers is thermal-aware job scheduling, i.e. job scheduling that relies on predictive thermal models to select among possible job schedules to minimize its energy needs. This paper investigates, using a more realistic linear cooling model, the energy savings of previously proposed thermal-aware job scheduling algorithms, which assume a less realistic model of constant cooling. The results show that the energy savings achieved are greater than the savings previously predicted. The contributions of this paper include: i) linear cooling models should be used in analysis and algorithm design, and ii) although the job scheduler must control the cooling equipment to realize most of the thermal-aware job schedule's savings, some savings can be still achieved without that control.

1 Introduction

Large data centers today contain up to tens of thousands of servers, consuming tens of megawatts of electricity annually [1, 2]. There is a growing problem of energy consumption that points toward seeking increasingly sophisticated ways, both in hardware and software, to achieve greater energy efficiency from data center facilities [3, 4, 5, 6, 7, 8]. Recent research has shown, through simulation, considerable savings for high-performance data centers through predictive *thermal-aware job scheduling*, i.e., scheduling that takes its thermal impact into consideration [7]. Previous work used a constant-value cooling model, i.e. cooling at a constant temperature, to estimate the energy savings. Nevertheless, most real-world cooling systems follow a discontinuous step-wise linear cooling model, i.e. cooling that supplies cool air at a temperature linearly dependent on that of the data center. This is pointed by both the technical specifications and *in situ* measurements, presented later in this paper.

This paper re-examines the energy savings of the XInt family [7, 9] of thermal-aware job scheduling algorithms under the step-wise linear cooling model. Specifically, it address the question "*what would the energy consumption of job schedules be in a step-wise linear cooling environment when they are derived by the XInt family algorithms under constant cooling by the XInt family algorithms under a step-wise linear cooling environment*." This is done by constructing an analytical temporal model of cooling and combining it with the heat recirculation as described by the *abstract linear heat interference model* [6] used by the XInt family. Numerical results show that energy savings

S. Ranka et al. (Eds.): IC3 2009, CCIS 40, pp. 568–580, 2009.

achieved are greater compared to previous results, while the order of energy savings of the examined algorithms is preserved.

The rest of the paper is organized as follows: Section 2 introduces concepts on data center layout, heat recirculation, cooling system efficiency, thermal maps and thermal-aware job scheduling. Section 3 presents the abstract linear heat recirculation model, and the XInt family of thermal-aware job scheduling algorithms that are evaluated. Section 4 introduces the cooling models, their thermal and power behavior. Section 5 presents simulation results of energy savings under the constant and linear cooling models. Section 6 concludes the paper.

2 Preliminaries

Data center Operation and layout. Figure 1 shows a typical data center's organization. Computing equipment sits on a raised floor plenum, organized into *rows* that separate *aisles*, alternating between cold air intake aisles and hot air exhaust aisles. Cold air from the *computer room air conditioner* (CRAC) is supplied to the room through grated tiles in the raised floor of cold aisles, to keep all servers below a manufacturer-specified *red-line* temperature. A data center is abstracted to consist of n nodes (chassis)[1]. Each node i consists of several processors (cores). Each node i draws air with inlet temperature T_i^{in}, adds heat by consuming power p_i and dissipates hotter air with outlet temperature T_i^{out}. The outlet temperature of a node comes from the combined activity of the servers in that node. The total computing power consumption of a data center is P^{comp}, that being the sum of all node power consumption: $P^{\text{comp}} = \sum_i p_i$. In the formulations in this paper, we use a *vectorized* notation for brevity, i.e. $\boldsymbol{p} = \{P_i\}_n$, $\boldsymbol{T}^{\text{in}} = \{T_i^{\text{in}}\}_n$ etc.

Heat recirculation and cooling efficiency. Contemporary data centers are cooled by chilled-water CRAC units, using conventional air-cooled methods (i.e. fan & heat sink) at the computing equipment. To keep all the equipment at a normal operating temperature, the CRAC has to supply cool air at an adequately low temperature. However, due to the non-linear cooling efficiency, the CRAC's set temperature affects the coefficient of performance (CoP). CoP characterizes the efficiency of heat removal; it is the ratio of the heat removed from a system over the work required to perform the removal. At any given point, the CRAC power can be described as [2]:

$$P^{\text{AC}} = \frac{P^{\text{in}}}{\text{CoP}\,(T^{\text{sup}})}. \tag{1}$$

where P^{in} is the heat rate of the air that enters the CRAC and $\text{CoP}(T^{\text{sup}})$ is the CoP of the cooling system when supplying cold air at T^{sup}.

Using the above equation in an equilibrium state, and assuming that all heat produced by computing equipment enters the CRAC, the total cost is the sum of the power used to run the equipment and the power used to cool it down:

$$P^{\text{total}} = P^{\text{comp}} + P^{\text{AC}} \overset{(1)}{=} \left(1 + \frac{1}{\text{CoP}(T^{\text{sup}})}\right) P^{\text{comp}}. \tag{2}$$

[1] As data centers use blade-based systems and power/ventilation of the blades is shared within each chassis, the abstractions here are chassis-oriented.

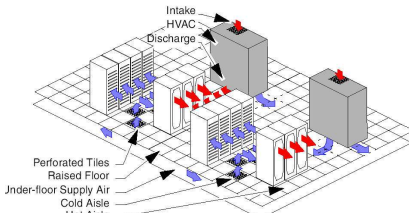

Fig. 1. Typical data center layout (source: ASHRAE [10])

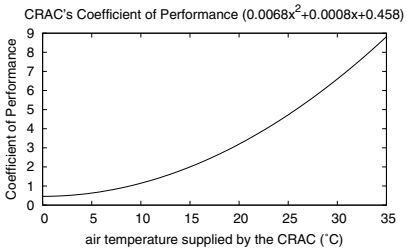

Fig. 2. Coefficient of performance of a typical CRAC unit. (source: [2])

Table 1. Scalar Symbols and Definitions

Symbol	Definition
n	number of server chassis
u_i	utilization of the i^{th} chassis
c_{ij}	temperature rise coefficient at j^{th} chassis caused by heat from i^{th} chassis.
$\boldsymbol{C}$	the matrix $\{c_{ij}\}$
p_i	power consumed by i^{th} chassis
a_i	power output coefficient of i^{th} chassis
b_i	idle power output of i^{th} chassis
q_j	number of cores requested by job j
$m, \dot{m}$	mass and mass flow rate, respectively
T_i^{in}	temperature of the i^{th} chassis
$\boldsymbol{T}^{\text{rise}}$	the vector of excess temperature above T^{sup} at the computing equipment inlets
T^{sen}	input air temperature of the CRAC
T^{sup}	output air temperature of the CRAC
T^{thres}	T^{sen} threshold between CRAC modes
P^{comp}	sum of the computing power
P^{out}	heat rate removed by CRAC(s)
P^{AC}	the power expended by CRAC(s)
α	slope of the line in linear cooling models
β	offset of the line in linear cooling models
ς	specific heat of air

Thermal Maps. Let $\boldsymbol{u} = \{u_1, u_2, \ldots, u_n\}$ be the utilization vector of n server chassis and $\boldsymbol{T}^{\text{in}} = \{T_1^{\text{in}}, T_2^{\text{in}}, \ldots, T_n^{\text{in}}\}$ be the corresponding stable-state temperature vector at the air inlets of these chassis for that utilization. The *thermal map* of the data center is defined as the translation function of $\boldsymbol{T}=F(\boldsymbol{u})$. This definition is a deterministic steady-state model. Such models are preferable in designing job scheduling algorithms mainly because they reduce the complexity of the decision making.

Thermal-aware job scheduling. The goal of thermal-aware scheduling is to allow the CRAC to act with higher efficiency by reducing the temperature it needs to supply to keep all compute equipment below its red-line temperature. The scheduler does so by allocating jobs to servers so that it reaches a *thermal balance* condition where all server inlet temperatures minimal and as equal as possible. As pointed out in previous research, ***the more balanced the inlet temperatures are, the higher the CRAC thermostat can be set at, thus saving more energy*** [7]. Much of the benefit of thermal-aware scheduling comes from dynamically setting the CRAC to maximize the CoP with each thermal map. One of this paper's contributions is that ***energy benefits can be still achieved without re-setting the CRAC***.

3 ALHI Model and XInt Job Scheduling Algorithms

Previous research on thermal-aware modeling and scheduling has produced a heat recirculation model, termed Abstract Linear Heat Interference model (ALHI), and the XInt

series spatio-temporal scheduling algorithms based on ALHI. This section will provide description of the ALHI and the XInt algorithms, whose performance will be evaluated in the simulation section.

3.1 The Abstract Linear Heat Interference Model

The ALHI asserts that the thermal map function F is linear [7]. The ALHI's core is a heat recirculation matrix $\boldsymbol{C}=\{c_{ij}\}_{n\times n}$, such that c_{ij} is the *temperature interference coefficient* of chassis i on the inlet temperature of chassis j, i.e., a heat rate of p_i will cause, by recirculation, a *temperature rise* of $c_{ij}p_i$ at the inlet of chassis j. Experimental results in the literature suggest a linear dependence of the power expended to the CPU utilization of the equipment [6, 7, 11]. Translating the power p_i to a server utilization rate u_i using a linear power model, we have $p_i = a_i u_i + b_i$, where b_i is the idle (i.e. zero utilization) power and a_i the linear coefficient of the utilization-to-power relation (a_i and b_i are usually obtained by power measurements). Inserting that to ALHI, we get the temperature rise vector:

$$\boldsymbol{T}^{\text{rise}} = \boldsymbol{C}\boldsymbol{p}^{\text{comp}} = \boldsymbol{C}(\boldsymbol{a} \odot \boldsymbol{u} + \boldsymbol{b}), \tag{3}$$

where $\odot$ is the element-wise product[2].

3.2 The XInt Algorithm Family

The algorithms are divided into *spatial-only*, which decide on the placement (i.e. the server assignment) only, and *spatio-temporal*, which decide on both the start time and placement of the jobs.

Spatial-only. The XInt scheduling algorithm uses Equation 3 to minimize the $\boldsymbol{T}^{\text{rise}}$. Equation 3 deals only with the spatial dimension of server utilization; as such, XInt [7] considers solving for the following job placement optimization problem:

> ***Given an idle data center, the recirculation matrix C, and the power parameters a and b, find a placement for a task of size*** q ***(i.e. requesting*** q ***cores) that minimizes*** $\max\{\boldsymbol{T}^{rise}\}$.

The XInt implementations used available software packages to solve this minimax formulation. The two methods used were a genetic algorithm (XInt-GA) and a sequential quadratic programming (XInt-SQP), the latter performed on the continuous version of the problem, and then regressing to a near integer solution [7]. Also, a variant of XInt was provided that performs thermal-aware placement of multiple jobs *simultaneously submitted* to a partially utilized data center [7].

An alternative to optimizing Equation 3 is to minimize $\sum\{T^{\text{rise}}\}$. Minimization of the summed $\boldsymbol{T}^{\text{rise}}$ is effectively minimization of the cumulative heat recirculation, as $\sum \boldsymbol{T}^{\text{rise}} = \sum \boldsymbol{C}\boldsymbol{p}^{\text{comp}} = \sum_i \sum_j c_{ij}p_i$. Using this optimization to perform thermal-aware job scheduling is based on the assertion that if the server with the minimum recirculated heat are selected to execute the jobs, then the $\boldsymbol{T}^{\text{rise}}$ vector will be reduced. The developed

[2] For example, $[a\ \ b\ \ c\ \ d] \odot [w\ \ x\ \ y\ \ z] = [aw\ \ bx\ \ cy\ \ dz]$.

least recirculated heat (LRH) heuristic, fully presented in [9], optimizes this formulation by pre-calculating *server ranks* as follows:

$$r_i = \sum_j v_j c_{ij} p_i^{\max}, \quad \text{where } v_j = \sum_i c_{ij} p_i^{\max},$$

where $p_i^{\max}$ is the maximum power output of node i. The algorithm then assigns tasks to the available (i.e. idle) servers with the minimal r_i values. The ranks v_j act as weight factors to the importance of the heat interference.

Spatio-temporal. The XInt algorithm above considers the spatial placement of the jobs only, with disregard to the duration of the jobs. New algorithms were developed to perform spatio-temporal scheduling and assess the energy savings of thermal-aware scheduling [9]. The spatio-temporal scheduling optimization problem examined was formulated as follows:

> ***Given an idle data center, the recirculation matrix C, and the power parameters a and b, find a spatio-temporal schedule for*** **a sequence of tasks** ***(each task*** i ***of size*** q_i***, arrival time*** t_i^{arr} ***and deadline*** t_i^{ded}***), in order to minimize the energy consumption.***

The algorithms developed are as follows:

FCFS-XInt: The FCFS-XInt is a combination of FCFS temporal placement (with back-filling), with XInt-based spatial placement. Benefits of this algorithm include its on-line nature and its high compatibility with FCFS-based commercial job schedulers.

SCINT: SCINT is a genetic algorithm (GA) implementation of spatio-temporal scheduling. It is an extension of XInt into the time dimension and it is an off-line algorithm. SCINT discretizes the time into ***time slots***, and using a GA approach constructs a schedule that resembles a slot-based server reservation table.

EDF-LRH: The SCINT and XInt algorithms are very slow for on-the-fly scheduling. MATLAB runs take several minutes for the SCINT and XInt-GA, and a few seconds for XInt-SQP. For that matter, a faster heuristic for the problem was developed that used an *earliest deadline first* (EDF) temporal scheduling with a *least recirculated heat* (LRH) spatial placement. The LRH optimization is defined as "minimize $\sum_{i=1}^{n} \boldsymbol{T}_i^{\text{rise}}$," which is a minimization problem, as opposed to a minimax problem such as XInt, and it takes a fraction of a second to compute [9].

4 Cooling Models

This section introduces three cooling models: a) the *constant* model, b) the segmented constant-linear model, and c) the stepwise linear model.

Constant Model. In this model, the CRAC provides a constant output temperature regardless of its input temperature (Figure 3a). Previous research routinely used the constant cooling model because it reaches a converging steady state solution in simulations [2, 7]. It is conceptually the simplest and fastest model to create a thermal map with. This model suppresses heat recirculation through the CRAC; thus it fails to capture any heat that would otherwise pass through the CRAC and not be extracted. This

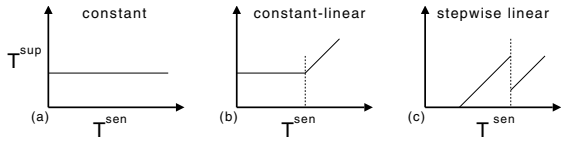

Fig. 3. Cooling models as characterized by their temperature transfer function: a) constant, b) segmented constant-linear, c) step-wise linear

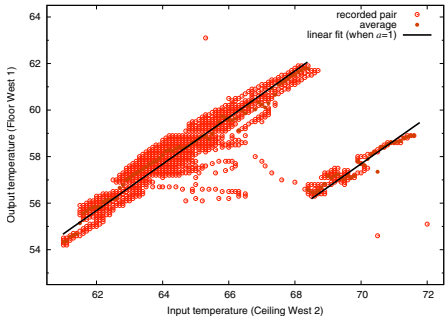

Fig. 4. Recorded pairs of input and output temperatures for an *in situ* CRAC unit, showing distinct operational modes

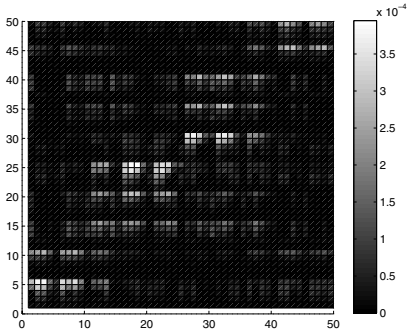

Fig. 5. The matrix C of coefficients as derived for the simulated data center in [7]

significantly alters the calculation of heat interference coefficients in a simulated environment with constant cooling model. It also suggests that the CRAC is capable of unboundedly extracting any heat—there is a maximum load a CRAC can cool. This drawback is addressed by the segmented constant-linear model, discussed in the following subsection.

Segmented constant-linear model. This model functions as the constant cooling model except that a maximum power load can be specified, beyond which all heat received is released back to the room (Figure 3b). It provides a constant temperature until the threshold temperature T^{thres} is reached and then follows a linear rise. This model is used in *computational fluid dynamics* (CFD) simulators to determine whether there exists sufficient cooling for a data center.

Stepwise Linear Model. The linear cooling model suggests that the temperature of the air within the CRAC is reduced in a linear fashion. It consists of linear segments $T^{\text{sup}} = \alpha T^{\text{sen}} + \beta$, where T^{sup} is the CRAC's output air temperature, and T^{sen} is the CRAC input air temperature (Figure 3c). This model has been realized from sensor measurements of *in situ* CRAC equipment, as shown in Figure 4.

Heat-extracting CRACs follow the stepwise linear model, where the CRAC switches between power modes according to the cooling needs, where in each mode the CRAC extracts a constant amount of heat, P^{out}. Each continuous section corresponds to a

separate P^{out}. For $P^{comp} > P^{out}$, the data center will keep heating up, while for $P^{comp} < P^{out}$, the data center will keep cooling down. The simulations presented in the next section will use this model to estimate the energy consumption of a data center.

5 Energy Consumption under Various Cooling Models

This section presents an evaluation of the schedulers' effective energy consumption under the constant and the step-wise linear models. The latter evaluation takes the schedules as calculated under the constant cooling assumption and examines them under the step-wise linear model. Effectively, the evaluation addresses the question "***what would the energy consumption of thermal-aware schedules be if they were produced assuming a constant cooling model but executed in an environment of step-wise linear cooling?***" The section first presents the results as derived under the constant cooling model, and then presents new results of the projected energy consumption of the same schedules using a step-wise linear model. Also, by considering portions where the computing power is constant, it also addresses the behavior of spatial-only placement algorithms.

5.1 Simulation Setup

Physical CFD modeling. FloVENT, a CFD simulator by Mentor Graphics, was used to obtain recirculation coefficient matrix used by the XInt, SCINT and LRH algorithms. A model of a data center with physical dimensions 9.6 m × 8.4 m × 3.6 m, was created in FloVENT. It 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 air flow rate 8 m^3/s. There are ten racks and each rack is equipped with five 7U (12.25-inch) chassis. The interference coefficient matrix in Figure 5 is derived from this setup. In this section, the *red-line* temperature is assumed at 35°C for all servers. The time slot length selected is 30 minutes.

System and job power profiles. In the simulations, we used 30 Dell PowerEdge 1955 chassis and 20 Dell PowerEdge 1855 chassis.

The algorithms' energy consumption has been evaluated using job traces from the ASU Fulton HPCI data center. The job traces provide: i) the job *arrival* times (i.e. the t_i^{arr}), ii) their corresponding *reservation* times, here treated as *deadlines* (i.e. the t_i^{ded}), iii) the number of servers required (i.e the q_i), and iv) the job start and finish times using the FCFS scheduling with back-filling. The estimates of the job execution times, t_i^{exe}, on the servers are based on the actual execution times in the ASU job traces, calculated as the difference between each job's start and finish times. The job traces are visualized in Figure 6, which shows the arrival time, estimated execution (both on the x-axis) and the number of processors requested (y-axis), and in Figure 7, which shows the arrival time, time reservation (treated as deadline), and the number of processors requested (y-axis).

The job execution time estimates for a type of node other than the one the job was actually run on, we simply multiply the execution time of the original equipment with the average gain in execution time on the other equipment. This gain is calculated as

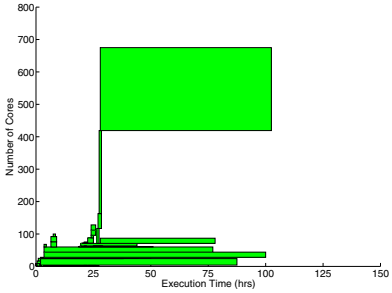

Fig. 6. Arrival, estimated execution time and number of processors of the job trace used in evaluating the scheduling algorithms [9]

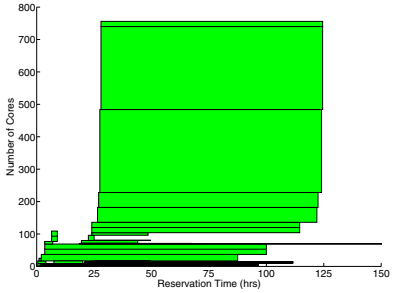

Fig. 7. Arrival, reservation time (in this paper it is treated as deadline) and number of processors of the job trace in Fig. 6

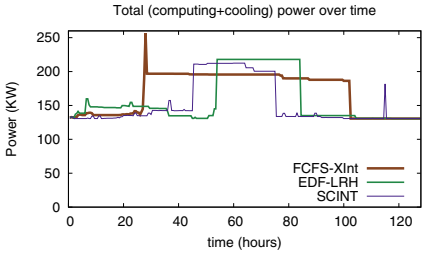

Fig. 8. Computing power consumption over time for the examined algorithms, under constant cooling, for the job trace in Figures 6,7 [9]

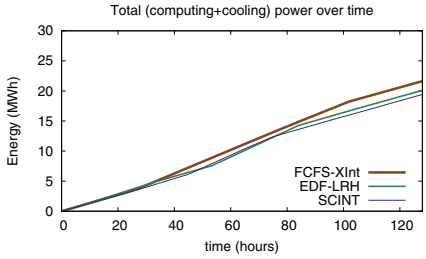

Fig. 9. Cumulative energy consumption (computing and cooling) over time for Figure 8, under the constant cooling

the ratio of the execution times between the two types of equipment measured[3] by Standard Performance Evaluation Corporation (SPEC) [12, 13]. From the estimates we get an average speed up of 2.5 when jobs run on 1955 servers in comparison to 1855 servers. The values of the power model used by FCFS-XInt, SCINT and EDF-LRH are: i) **PowerEdge 1855**: $a = 72$, $b = 1820$, ii) **PowerEdge 1955**: $a = 175$, $b = 2420$.

5.2 Energy Savings of under Constant Cooling Model

The ALHI model, as presented in [7, 9], was used in conjunction with a constant-temperature cooling supply. This means that the CRAC is supplying cool air at some constant temperature, T^{sup}, irrespective of the thermal condition of the data center. Therefore, the thermal map of the data center is analytically expressed as:

$$\boldsymbol{T}^{\text{in}} = \boldsymbol{T}^{\text{rise}} + T^{\text{sup}} = \boldsymbol{C}(\boldsymbol{a} \odot \boldsymbol{u} + \boldsymbol{b}) + T^{\text{sup}} \tag{4}$$

[3] The SPEC tests were run on both the Dell PowerEdge 1855 and 1955 servers using different benchmark applications, e.g. gzip, bzip, gcc, and so on.

This model asserts that were it not for interference, every chassis would have an inlet temperature equal to the CRAC temperature and the CRAC temperature could therefore be set to the lowest red-line temperature among the machines in the data center. Any CRAC output temperature configured below this value is the result of inefficiency caused by recirculation.

In-depth details on the simulation are given in [9]. The resulting total (computing and cooling) power consumption for the FCFS-XInt, EDF-LRH and SCINT are given in Figure 8. The per-chassis power break-down for each algorithm is given in Figures 10, 13, and 16.

5.3 Energy Consumption under the Step-Wise Linear Cooling Model

To calculate the energy consumption under the step-wise linear cooling model of the job schedules produced in the previous subsection, it is needed to predict the behavior of the CRAC given the schedule and the power equipment. For simplicity, we assume that the air coming out from the CRAC equally disperses into the room. For fixed P^{out} and P^{comp}, and for a data center air mass of m, we ***approximate*** the rate of temperature change at the input of the CRAC to be governed by the following equation:

$$T^{\text{sen}}(t) = T^{\text{init}} + \dot{T}^{\text{sen}}(t - t^{\text{sw}}) = T^{\text{init}} + \frac{-P^{\text{out}} + \sum_{i=1}^{\text{n}} \left(1 - \sum_{j=1}^{\text{n}} e_{ij}\right) p_i}{m\varsigma}(t - t^{\text{sw}}), \quad (5)$$

where $\{e_{ij}\}_{n \times n}$ is the power-to-power heat interference matrix (derived from $\boldsymbol{C}$, m and ς [7]), T^{init} is the starting temperature and t^{sw} is the time the CRAC takes to switch modes (here, it is assumed t^{sw}=10 minutes). The above equation asserts that the heat entering the CRAC is a linear weighted sum, based on the ALHI, of the servers' powers. The equation also accounts for a delay in switching between CRAC modes. If $\dot{m}$ is the CRAC's air mass flow rate, the temperature difference between T^{sen} and T^{sup} is:

$$T^{\text{sup}}(t) = T^{\text{sen}}(t) - \frac{P^{\text{out}}}{\dot{m}\varsigma}, \quad (6)$$

Methodology. Using Equations 5 and 6, we calculate the CRAC input and output temperatures. Then, using the temperature at the end of a time slot (as divided by SCINT) as the starting temperature of the next slot, we can calculate the T^{sup} over time. Using Equation 6 we can calculate the T^{sup}. Then, we can use the CoP to calculate the power consumption of the CRAC as:

$$P^{\text{AC}} = \frac{P^{\text{out}}}{\text{CoP}\,(T^{\text{sup}})} = \frac{P^{\text{out}}}{\text{CoP}\left(T^{\text{sen}} - \frac{P^{\text{out}}}{\dot{m}\varsigma}\right)}. \quad (7)$$

We use a three-mode, two-threshold CRAC with the following specifications:

$$\boldsymbol{P}^{\text{out}} = [5\,\text{W} \quad 75\,\text{W} \quad 250\,\text{W}], \quad \boldsymbol{T}^{\text{thres}} = [16\,^{\circ}\text{C} \quad 20\,^{\circ}\text{C}], \quad P^{\text{comp}} = \sum_{i=0}^{\text{n}} a_i u_i + b_i.$$

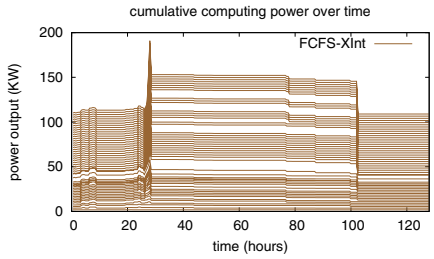

Fig. 10. Power output, divided per chassis, as yielded by FCFS-XInt's schedule

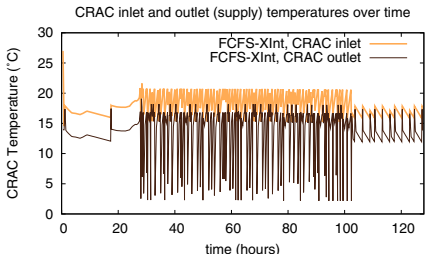

Fig. 11. CRAC input (T^{sen}) and output (T^{sup}) temperatures yielded for Figure 10

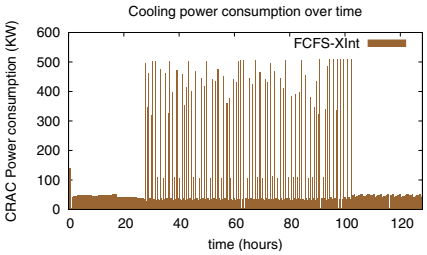

Fig. 12. CRAC consumed power for FCFS-XInt

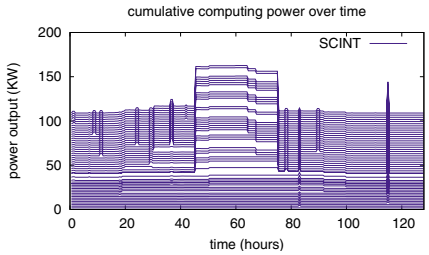

Fig. 13. Power output, divided per chassis, as yielded by SCINT's schedule

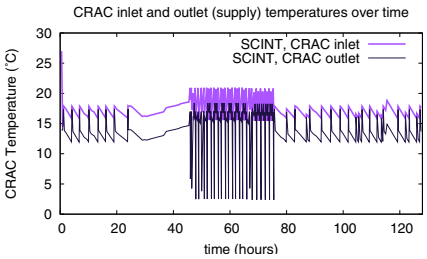

Fig. 14. CRAC input (T^{sen}) and output (T^{sup}) temperatures yielded for Figure 13

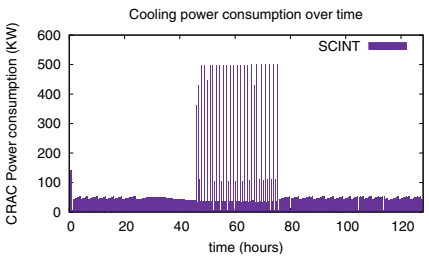

Fig. 15. CRAC consumed power for SCINT

Results. For each of the examined algorithms, we produced a set of figures consisting of 1) the server power graph, 2) the CRAC input and output temperatures, and 3) the CRAC power over the duration of the schedule. Figures 10, 13 and 16 show the power consumption of the produced schedules over time, divided into per-chassis lines (the wider the gap between two lines, the more power is output from that chassis at that moment); the topmost line is identical to the corresponding line in Fig. 8.

Figures 11, 14 and 17 show the resulting T^{sen} and T^{sup} temperatures over time, as calculated by Equations 5 and 6; we can see that the higher the heat rate generated, the faster and wider the oscillations among modes are.

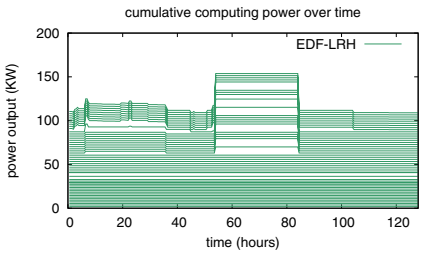

Fig. 16. Power output, divided per chassis, as yielded by EDF-LRH's schedule

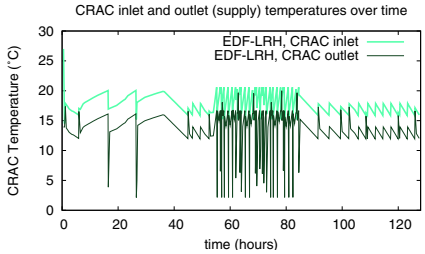

Fig. 17. CRAC input (T^{sen}) and output (T^{sup}) temperatures yielded for Figure 16

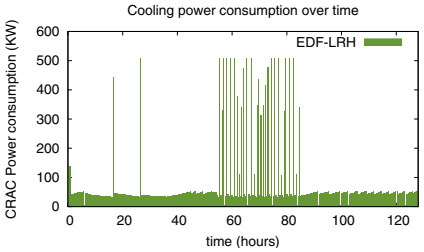

Fig. 18. CRAC consumed power for EDF-LRH

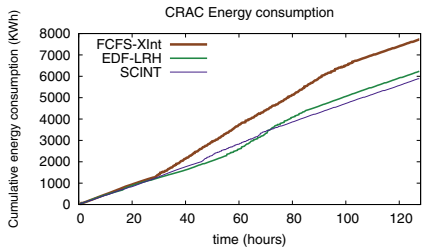

Fig. 19. Cumulative CRAC energy consumption from Figures 12, 15 and 18

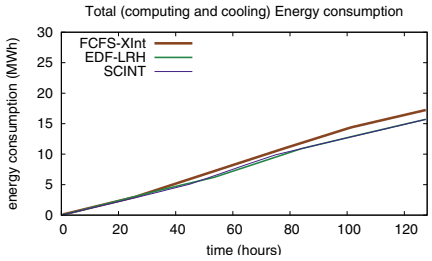

Fig. 20. Cumulative total (computing and cooling) energy consumption from Figures 8 and 19

Figures 12, 15 and 18 show the respective resulting power consumption over time presented as a filled curve (the surface corresponds to the cumulative energy), as calculated by Equation 7; it is clear that the FCFS-XInt algorithm causes a lot of high-power spikes to the CRACs.

Figure 19 provides the CRAC energy consumptions as they accumulate over time, for the three examined algorithms. Figure 20 provides the summed cumulative energy for computing and cooling, yielded from Figures 8 and 19.

6 Discussion and Conclusions

This paper introduced an analytical stepwise linear model to describe the behavior of CRAC systems in a realistic way. It also made analysis-based numerical estimations of the power consumption of schedules under the stepwise linear cooling model, produced by thermal-aware algorithms that assume a constant-value cooling model. The conclusions are summarized as follows:

Order of efficiency is preserved. One observation is that order of efficiency, as obtained in the constant cooling model, is preserved in the stepwise linear cooling model. Figure 19 shows that FCFS-XInt causes the worst cooling performance, followed by EDF-LRH and SCINT, which agrees with the constant-cooling results (Figure 8). Figure 20 shows the total (computing and cooling) energy as it accumulates over time.

Cooling oscillation seems to save energy. Comparing Figures 9 and 20, we see that the linear cooling model nominally reduces the total energy consumption; a preliminary explanation is that the oscillatory behavior of step-wise linear cooling in conjunction with the different power levels makes the CRAC conserve energy. This observation merits further investigation to be confirmed.

Savings can be achieved without re-setting the CRAC thermostat. In [7, 9], the optimizations rely on re-setting the CRAC thermostat to the highest allowed point, by solving Equation 4 for T^{sup} and replacing T^{in} with the red-line temperature. However, Figure 20 clearly shows that SCINT and EDF-LRH have lower energy consumption that FCFS-XInt, which means that energy savings are achieved without resetting the thermostat. We project that by re-setting the CRAC thermostat to appropriate levels over time, greater energy savings can be yielded. Therefore, realistic knowledge of the cooling system behavior helps create more accurate predictions and, in consequence, more efficient job schedules.

Acknowledgments

The authors thank the director of ASU's HPC Lab, Dan Stanzione, and his crew for granting access to ASU Fulton HPCI center, Mary Murphy-Hoye at Intel Corp. for donating the sensors to perform the measurements, and Michael Jonas for assisting into classifying and describing the cooling models. This work was partly funded by NSF grants #0649868 and #0834797, Intel Corp. and Science Foundation of Arizona.

References

1. Moore, J., Sharma, R., Shih, R., Chase, J., Patel, C., Ranganathan, P.: Going beyond CPUs: The potential of temperature-aware data center architectures. In: First Workshop on Temperature-Aware Computer Systems (June 2004)
2. Moore, J., Chase, J., Ranganathan, P., Sharma, R.: Making scheduling "cool": Temperature-aware resource assignment in data centers. In: 2005 Usenix Annual Technical Conference (April 2005)

3. Boucher, T.D., Auslander, D.M., Bash, C.E., Federspiel, C.C., Patel, C.D.: Viability of dynamic cooling control in a data center environment. ASME Journal of Electronic Packaging 128, 137–144 (2006)
4. Fontecchio, M.: Companies reuse data center waste heat to improve energy efficiency (May 2008)
5. Donald, J., Martonosi, M.: Techniques for multicore thermal management: Classification and new exploration. SIGARCH Comput. Archit. News 34(2), 78–88 (2006)
6. Tang, Q., Gupta, S.K.S., Stanzione, D., Cayton, P.: Thermal-aware task scheduling to minimize energy usage for blade servers. In: 2nd IEEE Int'l Dependable, Autonomic, and Secure Computing (DASC 2006) (September 2006)
7. Tang, Q., Varsamopoulos, G., Gupta, S.K.S.: Thermal-aware task scheduling for data centers through minimizing peak inlet temperature. IEEE Transactions on Parallel and Distributed Systems, Special Issue on Power-Aware Parallel and Distributed Systems (TPDS/PAPADS) 19(11), 1458–1472 (2008)
8. Heath, T., Diniz, B., Carrera, E.V., Meira, W.J., Bianchini, R.: Energy conservation in heterogeneous server clusters. In: Proceedings of the Symposium on Principles and Practice of Parallel Programming (PPoPP) (2005)
9. Mukherjee, T., Banerjee, A., Gupta, S.K.S.: Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers (Elsevier) Computer Networks, Special Issue on Resource Management in Heterogeneous Data Centers (accepted for publication) (2009)
10. ASHRAE: Thermal guidelines for data processing environments
11. Heath, T., Centeno, A.P., George, P., Ramos, L., Jaluria, Y.: 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, pp. 106–116. ACM Press, New York (2006)
12. SPEC: Standard Performance Evaluation Corporation – CINT2000 Result: Dell PowerEdge (1955), `http://www.spec.org/cpu2000/results/res2006q3/cpu2000-20060626-06297.html`
13. SPEC: Standard Performance Evaluation Corporation – CINT2000 Result: Dell PowerEdge (1855), `http://www.spec.org/cpu2000/results/res2005q3/cpu2000-20050902-04544.html`