Minimizing Data Center Cooling and Server Power Costs

Ehsan Pakbaznia and Massoud Pedram
University of Southern California
Los Angeles CA 90089
{*pakbazni*, *pedram*}*@usc.edu*

ABSTRACT

This paper focuses on power minimization in a data center accounting for both the information technology equipment and the air conditioning power usage. In particular we address the server consolidation (on/off state assignment) concurrently with the task assignment. We formulate the resulting optimization problem as an Integer Linear Programming problem and present a heuristic algorithm that solves it in polynomial time. Experimental results show an average of 13% power saving for different data center utilization rates compared to a baseline task assignment technique, which does not perform server consolidation.

ACM Categories & Subject Descriptors: Computer Systems Organization, Computer Applications

General Terms: Design, Management, Performance

Keywords: Data center, power dissipation, total cost of ownership

1. INTRODUCTION

Rapid increase in the World Wide Web (WWW) traffic is in part driven by a dramatic increase in requests to the popular web sites (social networking sites, online marketplaces), ubiquitous use of search engines, and web portals that combine media and entertainment, financial/market information, and email/chat service). Data centers form the center of the WWW and more broadly the cyber-universe. They sit at the heart of the information and communication technologies (ICT) ecosystem and have become essential to the functioning of business, service, academic, and governmental institutions.

Large data centers comprise of tens of thousands of servers with tens of peta bytes of storage, and multiple hundreds of giga bit bandwidth to the Internet. They typically serve millions of users globally and 24-7. Computing and storage capacities of these data centers are continually increasing; this is in turn made possible by advances in the underlying manufacturing process and design technologies available. A side effect of such a capacity increase has been a rapid rise in the energy consumption and power density of data centers. The electric bill of the data centers (including the electricity needed for cooling and air conditioning in the data center) is projected to pass 7 billion US dollars in the US alone, while the power density is expected to reach 60KW/m^2 for data centers by 2010. The Environmental Protection Agency (EPA), in its August 2007 report to the US Congress, affirmed that data centers consumed about 61 billion kilowatt-hours (kWh) in 2006, roughly 1.5 percent of total U.S. electricity consumption, for a total electricity cost of about $4.5 billion [1]. This level of electricity consumption is more than the electricity consumed by the nation's color TVs and similar to the amount of electricity consumed by approximately 5.8 million average U.S. households. The EPA report also stated that the energy consumption of servers and data centers has doubled in the past five years and is expected to quadruple in the next five years to more than 100 billion kWh at a cost of about $7.4 billion annually. If current trends continue, power demand of US data centers is expected to rise to 12 GW by 2011. According to a 2008 Gartner report [2], 50 percent of data centers will soon have insufficient power and cooling capacity to meet the demands of high-density equipment.

There are a number of different techniques currently employed to reduce the energy cost and power density in data centers. For example, load balancing [3][4] can be used to distribute the total workload of the data center among different servers evenly in order to balance the per server workload (and hence achieve uniform power density). Server *consolidation* [5], which refers to assigning incoming tasks to the minimum number of active servers in the data center and shutting down unused servers, is another approach for power reduction of data centers.

Accounting for about 30% of the total energy cost of a data center (another 10-15% is due to power distribution and conversion losses in the data center), the cooling cost is one of the major contributors of the total electricity bill of large data centers [6]. There have been several works attempting to reduce the energy required for cooling in a data center. The "hot-aisle/cold-aisle" structure, which has become common practice these days, is one of the attempts to improve the cooling efficiency of data centers (c.f. Section 2.1). There have been a number of prior work results on increasing the efficiency of the cooling process in data centers by performing temperature-aware task scheduling [7][8]. In [7] the authors present a heuristic approach that increases the cooling efficiency by minimizing a so called Heat Recirculation Factor. This reduces the amount of heat recirculation in the data center room and as a result, improves the cooling efficiency. The authors of [8] introduce three heuristic approaches to minimize the total power of a data center by scheduling tasks to have a uniform outlet temperature profile, minimum server power dissipation, or a uniform workload distribution, respectively.

Although the aforesaid approaches try to minimize the data center power consumption, they lack a precise objective function and/or accurate mathematical formulation of the optimization problem. Also, when it comes to the solution, they lack a rigorous algorithmic solution that solves the underlying optimization problem directly. For example, the uniform outlet temperature profile does not attempt to *minimize* the total data center power directly; instead it can be used as a technique to *reduce* the total data center power in an ad hoc manner. The authors of [8] formulate and solve a mathematical problem that maximizes the cooling efficiency of a data center. However, as we will see later in this paper, task assignment that maximizes the cooling efficiency without performing chassis consolidation does not result in minimum data center power dissipation.

We present a mathematical problem formulation for the total data center power optimization along with an efficient algorithm that minimizes the total data center power cost, i.e., server plus cooling power dissipations. The cooling power is reduced by choosing an optimum *supplied cold air* temperature value, T_s, whereas the server power is reduced by appropriately assigning incoming tasks to different servers and set the proper voltage-

frequency (*v-f*) level for each server depending on what type of task it is running. Task assignment and T_s selection are done as a result of solving a single mathematical problem.

The remainder of this paper is organized as follows. Section 2 presents the background information whereas Section 3 describes the data center power model used in this paper. Section 4 presents the optimization problem and our solution approach, respectively. Section 5 shows simulation results. Section 6 concludes the paper.

2. PRELIMINARIES

In this section we give an overview of a typical data center planning and arrangement of servers, hot/cold aisles, and the cooling system. Next we provide the data center power model, which is adopted in this paper. We also review the thermodynamic equations which are the basis for heat transfer and temperature distribution calculation in the data center.

2.1 DATA CENTER CONFIGURATION

A data center is typically a (warehouse-sized) facility with several rows of server racks. Each row comprises of several racks (cabinets), each rack contains several chassis, and each chassis contains multiple (Blade) servers. The servers can be single-core or multi-core processors. All blade servers in a chassis share a single power unit for the chassis.

A modern data center is designed in hot-aisle/cold-aisle style as depicted in Figure 1, where each row is sandwiched between a hot aisle and a cold aisle [5]. Cold air in the cold aisles is supplied by the air conditioning unit and comes through the perforated tiles in the floor underneath the cold aisles. Servers from different racks in adjacent rows suck the cold air coming from the cold aisle into the rack using chassis fans. The cold air cools the servers by carrying away the heat generated by these servers; the hot air exits the rack toward the adjacent hot aisles, and is then extracted from the room by the air conditioning intakes that are normally positioned on the ceiling above the hot aisles.

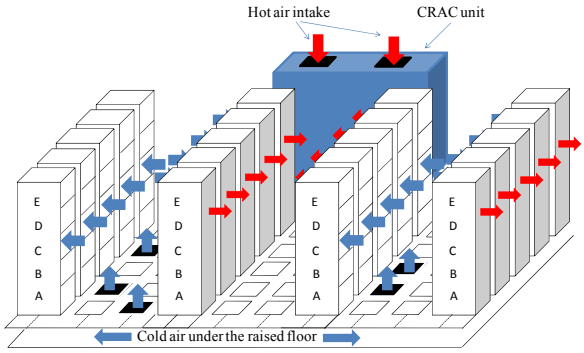

Figure 1. Hot-aisle/cold-aisle data center structure.

2.2 POWER MODEL FOR THE BLADE SERVERS

In this paper we assume that each task is assigned to only one server, and each server can only run at most one task. We also assume that dynamic voltage-frequency (*v-f*) scaling is available for the servers in a chassis i.e., servers can operate at different *v-f* levels depending on the type of the task that they are running.

Suppose the i^{th} chassis contains M_i servers. In addition, K+1 *v-f* levels are available to each server (including *v-f* = 0 corresponding to a fully power-gated server). Let w_{ij} denote the number of servers in the i^{th} chassis that are running at the j^{th} *v-f* setting. Evidently, $u_i \equiv \sum_{j=1}^{K} w_{ij} \leq M_i$ denotes the number of ON servers in the i^{th} chassis. The power consumption of a chassis may be calculated as:

$$P_i = \gamma_i + \alpha_i u_i + \sum_{j=1}^{K} \beta_{ij} w_{ij} \quad (1)$$

where γ_i, which represents the *base* power consumption of the i^{th} chassis, accounts for the power consumption of the chassis fan and switching losses due to AC-DC power conversion in the chassis. Similarly, α_i denotes the *uncore* power consumption of a server in the i^{th} chassis and represents power consumptions of the crossbar router, memory controller, I/O bridge, internal memory, and local hard drive in the i^{th} chassis. Finally, β_{ij} denotes the *core* power consumption and captures the power due to active core(s) and various caches in any server in the i^{th} chassis when the server is operating at the j^{th} *v-f* level. Notice that α_i does not scale with the *v-f* setting of the server whereas β_{ii} does.

The core power consumption of a server, i.e., β_{ii} in (1), depends on the type of the task that is running on the server. Tasks with lots of cache misses or high disk I/O accesses force the server to idle cycles, resulting in lower core power consumption. Conversely, tasks with low I/O accesses or cache miss rates tend to have higher server utilization, resulting in higher core power.

Although the utilization level of a server varies by the type of the task running on it, the difference between the power consumption of a fully-utilized server and, say, a 40%-60% utilized one is less than 2% for a typical blade server. For example, according to reference [8], for the Dell PowerEdge™ 1855 Blade Server [9] chassis with ten server slots, γ_i=820W, while the "uncore" power dissipation per blade server is 120W. Each server dissipates 50W and 30W of active power at 100% and 60% utilizations levels, respectively. The difference between power consumption values for different utilizations is small compared to the total power consumption of the chassis, and we can simply assume that the active power consumption of a server is largely independent of its utilization level. This assumption is also made by the authors of [10]. Due to high power consumption of an idle chassis, it is desirable to assign the incoming tasks to the minimum number of chassis so that the remaining ones can be turned off. This is called *chassis consolidation.*

Let N denote the number of chassis in the data center. We define $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_N]^T$ as a column vector representing base power dissipations of all chassis, and $\boldsymbol{\alpha}=[\alpha_i]_{N\times 1}$ and $\boldsymbol{\beta}=[\beta_{ij}]_{N\times K}$ as matrices capturing uncore and core power dissipations of all servers in the data center. $\boldsymbol{w}=[w_{ij}]_{N\times K}$ denotes the *server state matrix* where w_{ij} is the number of servers running at the j^{th} *v-f* level in the i^{th} chassis. Power consumption distribution of all chassis can be shown in vector form as $\boldsymbol{P} = [P_1, P_2, \ldots, P_N]^T$ and calculated as:

$$\boldsymbol{P} = \boldsymbol{\gamma} + \boldsymbol{\alpha} \odot \boldsymbol{u} + (\boldsymbol{\beta} \odot \boldsymbol{w}) \mathbf{1}_{K\times 1} \quad (2)$$

where $\mathbf{1}_{K\times 1}$ denotes a K-dimensional column vector with all elements equal to 1 and $\odot$ represents the element-by-element vector and matrix product operator, i.e., $\boldsymbol{\alpha} \odot \boldsymbol{u} = [\alpha_i u_i]_{N\times 1}$ and $\boldsymbol{\beta} \odot \boldsymbol{w} = [\beta_{ij} w_{ij}]_{N\times K}$.

2.3 HEAT TRANSFER EQUATIONS

The heat rate is defined as the amount of heat or thermal energy generated or transferred in a unit of time. The heat rate that is carried by an air flow is given by [12]:

$$Q = \rho f c_p T \quad (3)$$

where ρ is the air density in units of g/m^3, f is the air flow rate in units of m^3/s, c_p is the specific of the air in units of Jg^{-1}K^{-1}, T is the air temperature in units of Kelvin, and Q is the heat rate in Watts.

The temperature spatial granularity considered in this paper is at the chassis level. Each chassis draws the cold air to remove the heat from its hot servers. The hot air then exits the chassis from the

rear side. The temperature of the cold air that is drawn to the i^{th} chassis is called *inlet temperature* of that chassis and is denoted by T_{in}^{i}. Similarly, the *outlet temperature* of the i^{th} chassis, T_{out}^{i}, is defined as temperature of the hot air that exits the chassis. Consider the i^{th} chassis with a power dissipation of P_i, inlet and outlet temperatures of T_{in}^{i} and T_{out}^{i}, and an air flow rate of f_i. From the law of energy conservation, we can write:

$$Q_{in} + P_i = Q_{out} \Rightarrow P_i = \rho f_i c_p \left(T_{out}^{i} - T_{in}^{i}\right) \quad (4)$$

Given a power distribution among the chassis in the room, and given a fixed condition for the Computer Room Air Conditioning (CRAC) unit, e.g., fixed temperature and flow rate for the supplied cold air, if we wait for a long enough time, we reach a steady state condition for the temperature profile in the room. The steady state temperature distribution in the room is determined by the inlet and outlet temperatures of different chassis. The inlet temperature of a chassis depends on the supplied cold air from the CRAC and the hot air that is *re-circulated* from the outlet of other chassis. The outlet temperature of a chassis depends on the inlet temperature and the power consumption of that chassis.

An abstract heat model has been proposed for data centers in [9] where the authors show that the recirculation of heat in a data center can be described by a cross-interference matrix. The cross-interference matrix is represented by $\Phi_{N\times N} = \{\phi_{ij}\}$ and shows how much of the inlet heat rate of each chassis is contributed by the outlet heat rate of other chassis. More precisely, ϕ_{ij} shows the contribution of the outlet heat rate of the i^{th} chassis in the inlet heat rate of the j^{th} one. If Q_{out}^{i} and Q_{in}^{j} denote the outlet and inlet heat rates for the i^{th} and j^{th} chassis respectively, the inlet heat rate for different chassis may be calculated as follows [11]:

$$Q_{in}^{j} = \sum_{i=1}^{n} \phi_{ij} Q_{out}^{i} + Q_s + P_j \quad j = 1, 2, \dots, N \quad (5)$$

In the vector form, we can write:

$$\boldsymbol{Q}_{in} = \boldsymbol{\Phi}^T \boldsymbol{Q}_{out} + \boldsymbol{Q}_s + \boldsymbol{P} \quad (6)$$

where, $\boldsymbol{Q}_{in} = \left[Q_{in}^{1}, \dots, Q_{in}^{N}\right]^T$, $\boldsymbol{Q}_{out} = [Q_{out}^{1}, \dots, Q_{out}^{N}]^T$, $\boldsymbol{P} = [P_1, \dots, P_N]^T$, and $\boldsymbol{Q}_s$ is a column vector of size N with all entries set to $Q_s = \rho f c_p T_s$. The heat rate can be transformed to the temperature using thermodynamic constants [11]:

$$\boldsymbol{T}_{in} = \boldsymbol{T}_s + \boldsymbol{D}\boldsymbol{P}, \quad \boldsymbol{D} = [(\boldsymbol{K} - \boldsymbol{\Phi}^T \boldsymbol{K})^{-1} - \boldsymbol{K}^{-1}] \quad (7)$$

where $\boldsymbol{T}_{in}$ and $\boldsymbol{T}_s$ are the corresponding inlet temperature and the cold air supply vectors, respectively and $\boldsymbol{K}$ is an N×N diagonal matrix whose entries are the thermodynamic constants of different chassis, i.e., $\boldsymbol{K} = diag(K_1, \dots, K_N)$, and $K_i = \rho f_i c_p$. It is clear from (7) that the power distribution among different chassis in the data center directly affects the temperature distribution in the room. If we use equation (2) to substitute $\boldsymbol{P}$ into (7), we have:

$$\boldsymbol{T}_{in} = \boldsymbol{T}_s + \boldsymbol{D}(\boldsymbol{\gamma} + \boldsymbol{\alpha} \odot \boldsymbol{u} + (\boldsymbol{\beta} \odot \boldsymbol{w})\boldsymbol{1}_{K\times 1}) \quad (8)$$

3. Data Center Power Modeling

3.1 Power Consumption of the CRAC Unit

The cooling process of a data center is performed by the CRAC unit (c.f. Section 2.1) where the hot air transfers its heat to some cold substance, usually cold water or air, while it is being passed through a pipe in the CRAC unit. When it is cold enough, the air is ready to enter the room using the CRAC fans. The heated substance in turn goes to a chiller to get cold again.

The efficiency of the cooling process depends on different factors such as the substance used in the chiller, the speed of the air exiting the CRAC unit, etc. *Coefficient of Performance* (COP), which is a term used to measure the efficiency of a CRAC unit, is defined as the ratio of the amount of heat that is removed by the CRAC unit (Q) to the total amount of energy that is consumed in the CRAC unit to chill the air (E) i.e., [7]:

$$COP = Q/E \quad (9)$$

The COP of a CRAC unit is not constant and varies by the temperature of the cold air that it supplies to the room. In particular the higher the supplied air temperature, the better cooling efficiency. In this paper we use the COP model of a typical water-chilled CRAC unit which has been utilized in a HP Utility Data Center [7]. This model is quantified in terms of the supplied cold air temperature (T_s) as follows [7]:

$$COP(T_s) = (0.0068\, T_s^2 + 0.0008\, T_s + 0.458) \quad (10)$$

3.2 Total Power Consumption

By the total power dissipation of a data center, we mean the summation of the power consumptions of all chassis and the CRAC unit i.e., we do not consider power losses in the electrical power conversion network (UPS, AC-DC and DC-DC converters) as well as losses in the switch gear and conductors. This component of power consumption in a well-designed data center is typically equal to a fixed percentage of the power consumption of the information technology (IT) equipment and CRAC unit.

The IT power consumption of a data center is denoted by P_{IT} and is the summation of power consumption over all chassis:

$$P_{IT} = \sum_{i=1}^{N} P_i \quad (11)$$

where P_i is the power consumption in the i^{th} chassis and N is the total number of chassis in the data center. From reference [6], the power cost of the CRAC unit may be specified as[7]:

$$P_{CRAC} = \frac{P_{IT}}{COP(T_s)} \quad (12)$$

The total data center power consumption is the summation of P_{IT} and P_{CRAC} and can be written as:

$$P_{DC} = \left(1 + \frac{1}{COP(T_s)}\right) \sum_{i=1}^{N} P_i \quad (13)$$

Substituting the expression from (1) for P_i, we obtain:

$$P_{DC} = \left(1 + \frac{1}{COP(T_s)}\right) \left(\sum_{i=1}^{N} \gamma_i + \sum_{i=1}^{N} \alpha_i u_i + \sum_{i=1}^{N} \sum_{j=1}^{K} \beta_{ij} w_{ij}\right)$$

4. Data Center Power Minimization

4.1 Problem Statement

We consider the steady state data center problem where the tasks run for a long period of time on the servers in a data center, resulting in stationary temperature profile. Clearly, a change in the power dissipations of any subsets of servers will affect the temperature distribution in the room. It takes order of minutes for the temperature to reach its new steady state distribution [13]. If execution times of the tasks are long, the steady state assumption will be valid because a change in the workload and server state matrix takes place at a timing granularity which is longer than the time needed to reach the steady state temperature distribution in the data center. This is usually the case for the High Performance Computing (HPC) scenarios where tasks can run for hours or even days on the servers. For example, consider the tasks that have to be executed in the course of designing a new VLSI chip, synthesis, timing analysis, place and route, HSPICE simulation, etc. These are all the tasks that are usually run on one or more servers for a long period of time. Other examples include scientific computation

for weather prediction, financial analysis, physics-based simulation, multi-player virtual world games, etc.

We assume that the desired *v-f* setting for each task is determined from the service level agreement and/or domain name servers (DNS) of the requester. For tasks with unconstrained or relaxed turnaround times, we can simply set the *v-f* level to the lowest possible level to save power. For tasks with stringent turnaround times, the corresponding *v-f* level can be determined by performing a linear mapping from the requested turnaround time to an appropriate *v-f* level. In the case of HPC tasks that are required to be executed as fast as possible, the *v-f* level can be determined based on workload characteristics. For example, for HPC tasks with low instruction per cycle (IPC) and high cache miss rate (CMR) values where the server will be mostly awaiting data to be fetched from the L2 cache or off-chip memory, using higher *v-f* level for the server will have little impact on the actual server performance, e.g., instructions per second (IPS), of the processor while it will increase the energy dissipation. Thus, the *v-f* level mapping for tasks can be done by prior (historic) knowledge about workload characteristics. A detailed discussion of how exactly to do this is outside the scope of this paper.

Our goal is to minimize the total data center power consumption given in (13) by (i) determining the optimum value of T_s, (ii) turning various servers and chassis ON/OFF, and (iii) for ON chassis determining the number of the ON servers their corresponding cores' *v-f* levels. Let the required number of servers to serve a given set of tasks be S_{tot}. The power optimization problem for serving the given set of tasks is as follows:

$$\text{Minimize}\left\{\left(1+\frac{1}{\text{COP}(T_s)}\right)\sum_{i=1}^{N} P_i\right\}$$

s. t.

1. $\mathbf{T}_{in} \le \mathbf{T}_{critical}$
2. $0 \le u_i \le M_i \quad ; \quad i = 1,2,\dots,N$
3. $u_i = \sum_{j=1}^{K} w_{ij} \quad ; \quad i = 1,2,\dots,N$
4. $\sum_{i=1}^{N} w_{ij} = S_j \quad ; \quad j = 1,2,\dots,K$

(14)

where $\mathbf{T}_{in}$ denotes the inlet temperature vector of the chassis, and $\mathbf{T}_{critical}$ is a vector of size N with all entries equal to the critical inlet temperature, $T_{critical}$ (The inlet temperature of all chassis must be less than this value in order to ensure that the corresponding servers will not overheat and eventually fail). A typical value for $T_{critical}$ is 25°C [11]. S_j is the total number of required servers with the j^{th} *v-f* setting. Clearly, $S_{tot} = \sum_{j=1}^{K} S_j$.

4.2 Problem Re-Statement

For simplicity, we assume that the data center is initially idle, that is, all servers and chassis in the data center are assumed to be OFF initially. First separate T_s from the cost function in (14) i.e., determining the optimum value of T_s later. For a fixed value of T_s, $COP(T_s)$ becomes a constant and can thus be taken out of the cost function. In this case, the cost function simply becomes P_{IT}:

$$\text{Minimize}\left\{\sum_{i=1}^{N} P_i\right\} \tag{15}$$

Next we define an integer variable for each chassis that takes on values from {0,1} and signals whether a chassis is ON or OFF. This variable for the i^{th} chassis is denoted by X_i:

$$X_i = \begin{cases} 0 \; ; \; u_i = 0 \\ 1 \; ; \; u_i \neq 0 \end{cases} \tag{16}$$

The power consumption of the i^{th} chassis with $u_i = \sum_{j=1}^{K} w_{ij}$ ON servers can be rewritten as:

$$P_i = X_i\left(\gamma_i + \alpha_i u_i + \sum_{j=1}^{K} \beta_{ij} w_{ij}\right) \tag{17}$$

Noting $X_i w_{ij} = w_{ij}$ and $X_i u_i = u_i$, the cost function in (15) becomes:

$$\text{Minimize}\left\{\sum_{i=1}^{N} \gamma_i X_i + \sum_{i=1}^{N} \alpha_i u_i + \sum_{i=1}^{N}\sum_{j=1}^{K} \beta_{ij} w_{ij}\right\} \tag{18}$$

We can write (18) in an equivalent matrix form as follows:

$$\text{Minimize}\{\boldsymbol{\gamma}^T \boldsymbol{X} + \boldsymbol{\alpha}^T \boldsymbol{u} + \text{tr}(\boldsymbol{\beta}^T \boldsymbol{w})\} \tag{19}$$

where $\boldsymbol{X}=[X_1,X_2,\dots,X_N]^T$, and *tr*(.) is the *trace* operation defined on square matrices. With this new notion, the power vector and the inlet temperature vector in (8) will change to:

$$\boldsymbol{P} = \boldsymbol{\Gamma X} + \boldsymbol{\alpha} \odot \boldsymbol{u} + (\boldsymbol{\beta} \odot \boldsymbol{w})\mathbf{1}_{K\times 1} \tag{20}$$

$$\boldsymbol{T}_{in} = \boldsymbol{T}_s + \boldsymbol{D}(\boldsymbol{\Gamma X} + \boldsymbol{\alpha} \odot \boldsymbol{u} + (\boldsymbol{\beta} \odot \boldsymbol{w})\mathbf{1}_{K\times 1}) \tag{21}$$

where $\boldsymbol{\Gamma}$ is a diagonal matrix defined as $\boldsymbol{\Gamma} = diag(\gamma_1, \gamma_2, \dots, \gamma_N)$. Changing $\boldsymbol{\gamma}$ to $\boldsymbol{\Gamma X}$ makes the base power components of the OFF chassis to disappear from the temperature equation.

Next we show that the problem of minimizing (19) with the constraints given in (14), can be formulated and solved as an *Integer Linear Programming* (ILP) problem. We stress that although the relationship between each X_i and its corresponding u_i which is expressed in (16) is not a linear relationship, we show that it can be replaced by equivalent linear constraints.

Lemma 1 The nonlinear relationship shown in (16) holds for $X_i \in \{0,1\}$ and $u_i \in \{0,1,\dots,M_i\}$ *if and only if* the following linear inequalities are satisfied:

$$X_i \le u_i \le M_i X_i \tag{22}$$

Proof: First assume that (16) holds. If $u_i = 0$, $X_i = 0$ and both of the inequalities in (22) hold. If $u_i \neq 0$, $X_i = 1$ and we have $1 \le u_i \le M_i$ which is equivalent to $X_i \le u_i \le M_i X_i$. To prove the reverse direction we use contradiction. Assume that the inequalities in (22) hold, but (16) does not. One of the following scenarios can happen: $u_i = 0$ and $X_i = 1$ or $u_i \neq 0$ and $X_i = 0$. If $u_i = 0$ and $X_i = 1$, by substituting X_i and u_i values in (22), we have: $1 \le 0 \le M_i$ which is clearly false. On the other hand, if $u_i \neq 0$ and $X_i = 0$, substituting X_i and u_i values in (22) will result in $0 \le u_i \le 0$ which means $u_i = 0$, but this contradicts with our assumption of $u_i \neq 0$. Therefore, (22) has to hold when (16) holds. ■

Theorem 1 The optimization problem given in (14) with a fixed T_s can be formulated as the ILP problem given in (23) where $\boldsymbol{X}=[X_1,X_2,\dots,X_N]^T$ and $\boldsymbol{w}=[w_{ij}]_{N\times K}$ are the unknowns.

Proof: Proof is straightforward using Lemma 1, linear power model in (1), and equation (21) for $\boldsymbol{T_{in}}$. ■

$$\text{Minimize}\{\boldsymbol{\gamma}^T \boldsymbol{X} + \boldsymbol{\alpha}^T \boldsymbol{u} + \text{tr}(\boldsymbol{\beta}^T \boldsymbol{w})\}$$

s. t.

1. $\boldsymbol{T}_s + \boldsymbol{D}(\boldsymbol{\Gamma X} + \boldsymbol{\alpha} \odot \boldsymbol{u} + (\boldsymbol{\beta} \odot \boldsymbol{w})\mathbf{1}_{K\times 1}) \le \mathbf{T}_{critical}$
2. $X_i \le u_i \le M_i X_i \quad ; \quad i = 1,2,\dots,N$
3. $u_i = \sum_{j=1}^{K} w_{ij} \quad ; \quad i = 1,2,\dots,N$
4. $\sum_{i=1}^{N} w_{ij} = S_j \quad ; \quad j = 1,2,\dots,K$
5. $X_i \in \{0,1\} \quad ; \quad i = 1,2,\dots,N$
6. $w_{ij} \in \{0,1,\dots,M_i\} \quad ; \quad i = 1,2,\dots,N \; ; \; j = 1,2,\dots,K$

(23)

4.3 SOLUTION TO THE STEADY STATE PROBLEM

The ILP problem given in (23) can be solved using the well known ILP solver packages or it can be solved heuristically as explained next. Figure 2 shows a proposed algorithm to solve the power minimization problem in (14). The algorithm is called *MINTOTDATACENTERPOW,* or *MTDP*(.) for short. Input to this algorithm is the number of incoming tasks at different *v-f* levels, S_j's. The algorithm varies T_s from MIN_T_s to MAX_T_s and solves the ILP optimization problem in (23) for each T_s value. Feasible solutions for different T_s values are compared and the one with the minimum total power, i.e., s_{min}, is returned as the solution.

Algorithm: $MTDP(C_1, C_2, \dots, C_K)$

```
1:  /* Total data center power minimization*/
2:  T_s = MAX_T_s;
3:  while (T_s ≥ MIN_T_s) do
4:      (w,X) = CONSOLIDATION LP (T_s, C_1, ..., C_K);
5:      s = (w,X);
6:      ŝ = FIND FEASIBLE SOLN (s);
7:      if {ŝ} ≠ {};
8:          S = S ∪ {(ŝ, T_s)}
9:      end if
10:         T_s = T_s – Δ
11: end while
12: s_min = Min (S)
13: return s_min
```

Figure 2. Top-level algorithm to solve problem in (14).

Algorithm: *FIND FEASIBLE SOLN* (***w, X***)

```
1:  /* Find the closest integer solution to LP*/
2:  for (i,j =1 to i = N, j=K) do
3:      w_ij = round(w_ij);
4:  end for
5:  for (i = 1 to i = N) do
6:      if (X_i > ½) X_i = 1;
7:  end for
8:  while (∃ i | X_i ∉ ℕ_0) do
9:      u = argmin_{1≤i≤N, X_i∉ℕ_0, X_i≤0.5} {X_i P_i(w)};
10:     while (∃ j | w_uj ≠ 0) do
11:         if (∃ r | 0 < Σ_k w_rk < M) then
12:             v = argmin_{1≤r≤N} {Σ_{i=1}^{N} d_ir};
13:             w_uj = w_uj – 1; w_vj = w_vj + 1;
14:         else X_u=1;
15:     end while
16: end while
17: T_s = T_crit – DP(w);
18: return (T_s,w,X)
```

Figure 3. Converting LP solution to the closest ILP solution for (23).

We first remove integer constraints 4 and 5 in (23) and solve the resultant *Linear Programming* (LP) version of the problem by using *CONSOLIDATIONLP*(.) routine which is a regular LP solver. Next we pass the LP result to *FINDFEASIBLESOLN*(.), described in Figure 3, which finds the closest ILP solution. This heuristic essentially takes the solution to the LP problem and finds an ILP solution with the total power value close to the LP power value. This is done by attempting to turn OFF chassis with X_i < ½ and distributing their workload among the already ON chassis. If all the tasks on a chassis are distributed among other ON chassis successfully, the heuristic then turns this chassis OFF by setting its corresponding *X* value to 0. Note that second to the fifth constraints in (23) are already taken care of during the proposed heuristic and we make sure that the first constraint is also satisfied by adjusting T_s at the end. It is easy to show that the time complexity of this heuristic algorithm is $O(N^2 S_{tot} \log N)$.

5. SIMULATION RESULTS

We compare power dissipation and cooling cost of our proposed algorithm, called MTDP, with a baseline comprised of the task assignment algorithm of [8] (Xint_SQP) augmented by an automatic shutdown mechanism for all idle servers and chassis. This baseline algorithm is denoted by BASELINE in the results. Note that Xint_SQP maximizes the cooling efficiency by assigning the tasks in such a way that lowers the peak inlet temperature of chassis. This also maximizes the required supplied cold air temperature value (T_s), resulting in more efficient cooling due to higher COP(T_s) value. BASELINE uses the same task assignment algorithm as Xint_SQP. In addition, it turns off idle servers and chassis to create a more competitive baseline algorithm for comparison with our proposed solution, MTDP.

We use MATLAB to perform our simulations in this paper. In order to solve the ILP problem we used TOMLAB [14], an ILP solver package for MATLAB. A small scale data center with physical dimensions of 9.6m×8.4m×3.6m consisting of 7U blade type servers similar to the one used in [8] has been used in this paper. The data center has two rows that are put together as hot-aisle/cold-aisle arrangement. Each row has five 42U racks. Each rack consists of five chassis each having 20 single-core servers. Therefore, there are a total number of 1000 servers in the data center. A CRAC unit is used to supply the cold air with f=8m^3/s in the room. Two, three including *v-f* =0, different *v-f* levels (K=2) are available in our simulations, VF1-VF2 with VF1 representing the higher supply voltage and frequency levels. We assume a homogeneous data center where different chassis (servers) have similar power/performance characteristics. Power parameters of servers and chassis used in this section are shown in Table 1.

Table 1. Power parameters used in simulations.

Voltage-Frequency Level	Chassis Overhead (γ) (W)	Uncore Power of Server (α_i) (W)	Core Power Dissipation (β_{ij}) (W)
VF1	820	60	25
VF2	820	60	12.5

We compare the total data center power dissipation, including chassis and cooling costs, resulting from MTDP and BASELINE. Results are reported for different data center utilizations, capturing percentage of the active servers. Since each active server runs only one task, the data center utilization is varied by varying number of the tasks that are being assigned to the data center, e.g., to simulate a 30% utilized data center, we assign 300 tasks (there are a total of 1,000 servers) to the data center.

Figure 4 shows the total power dissipation values for both BASELINE and MTDP solutions. Since BASELINE does not support dynamic *v-f* scaling, for these results, we set all ON servers to VF1 for both BASELINE and MTDP. It can be seen from Figure 4 that the power dissipation of MTDP is always less than that of BASELINE for different data center utilization values except for the full utilization case where both solutions produce the

same results because all servers must be ON, and hence, chassis consolidation plays no role in reducing the data center power. The reason that MTDP performs better than BASELINE is that BASELINE does not perform chassis consolidation, and hence, it cannot minimize the total data center power.

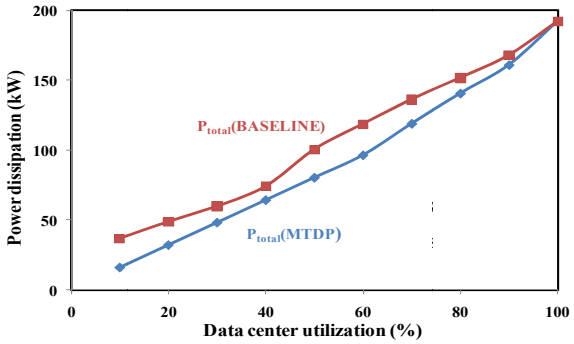

Figure 4. Comparing the power dissipation of MTDP and BASELINE algorithms.

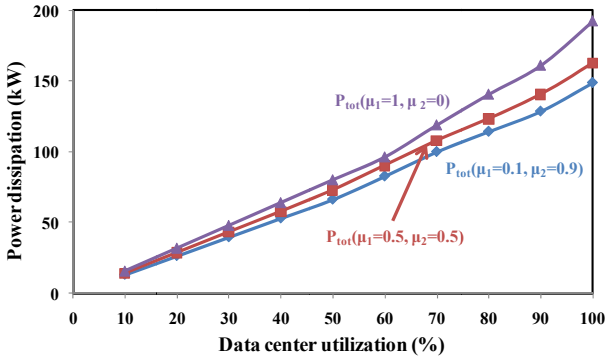

Figure 5. Comparing three different workload population scenarios for their total data center power dissipation.

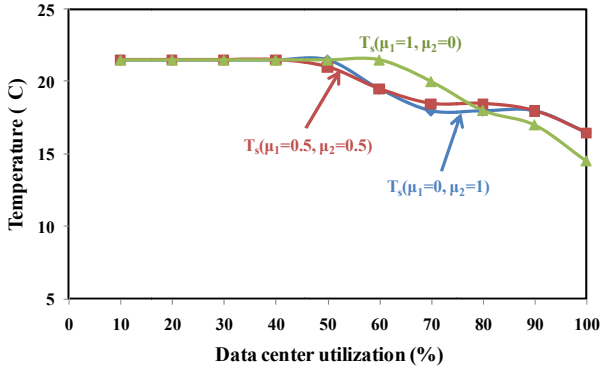

Figure 6. Comparing three different workload population scenarios in terms of their supplied cold air temperatures.

We run our proposed algorithm to schedule a given set of tasks with a given workload population to see the resulting temperature distribution in the data center. Figure 5 and Figure 6 show the total power dissipation and T_s value, respectively, versus the data center utilization for different workload populations. We set K=2, i.e., two active *v-f* levels plus *v-f* = 0, and defined two coefficients, γ_1 and γ_2, where γ_j denotes the ratio of S_j to S_{tot}, to distinguish between workloads based on their required performance level. Figure 7 shows the temperature distribution of different chassis in the example data center after assigning a workload with 60% data center utilization and workload population with μ_1=0.1 and μ_2=0.9.

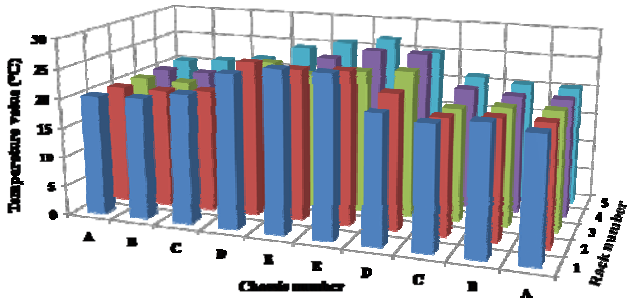

Figure 7. Steady state temperature distribution in the data center with 600 ON servers, μ_1=0.1 and μ_2=0.9.

6. CONCLUSION

We formulated and solved an optimization problem to minimize the total data center power dissipation using a thermal-aware task assignment technique which allocates the tasks on different servers in the data center and assigns a *v-f* setting for each selected server. The proposed algorithm returns the optimum cold air supply temperature, task assignment on different servers, and the corresponding *v-f* level setting. The proposed technique resulted in high power savings under different data center utilizations. Experimental results showed average of 13% power saving for different data center utilization values compared to a baseline task assignment technique that does not perform chassis consolidation.

REFERENCES

[1] "Report to congress on server and data center energy efficiency," *U.S. Environmental Protection Agency*, Aug. 2007.

[2] "Meeting the DC power and cooling challenge," *Gartner*, 2008.

[3] V. Cardellini, et al. "Dynamic load balancing on Web-server systems," *IEEE Internet Computing Magazine*, vol. 3, issue 3, May/June 1999, pp. 28-39.

[4] D. M. Dias et al. "A scalable and highly available web server," *Proc. IEEE Computer Soc. Int'l Conf.,* Feb. 1996, pp. 85–92.

[5] http://www.sun.com/x64/intel/consolidate-using-quadcore.pdf

[6] N. Rasmussen, "Calculating total cooling requirements for data centers," *American Power Conversion*, White Paper #25, 2007.

[7] J. Moore, et al. "Making scheduling 'cool': Temperature-aware resource assignment in data centers," *Usenix Annual Technical Conf.*, Apr. 2005.

[8] Q. Tang, et al. "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach," *IEEE Trans. on Parallel and Distributed Systems*, vol. 19, issue 11, Nov. 2008, pp.1458-1472.

[9] http://www.newservers.com/1855-specs.pdf

[10] E. Pinheiro, et al. "Load balancing and unbalancing for power and performance in cluster-based systems," *Proc. Workshop on Compilers and Operating Systems for Low Power,* 2001.

[11] Q. Tang, et al. "Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters," *Proc. Int'l Conf. on Intelligent Sensing and Info. Process.,* Dec. 2006, pp. 203-208.

[12] Y. A. Cengel, *Heat transfer: a practical approach*, 2nd edition, McGraw-Hill, 2003.

[13] J. Moore, et al. "Going beyond cpu's: the potential of temperature-aware data center architectures," *Proc. Workshop on Temperature-Aware Computer Systems*, Jun. 2004.

[14] http://tomopt.com/tomlab/