THE INSTITUTE FOR SYSTEMS RESEARCH

ISR TECHNICAL REPORT 2010-8

Unified Datacenter Power Management Considering On-Chip and Air Temperature Constraints

Bing Shi, Ankur Srivastava

ISR develops, applies and teaches advanced methodologies of design and analysis to solve complex, hierarchical, heterogeneous and dynamic problems of engineering technology and systems for industry and government.

ISR is a permanent institute of the University of Maryland, within the A. James Clark School of Engineering. It is a graduated National Science Foundation Engineering Research Center.

www.isr.umd.edu

Unified Datacenter Power Management Considering On-Chip and Air Temperature Constraints

Bing Shi and Ankur Srivastava
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742, USA
e-mail:{bingshi,ankurs}@umd.edu

Abstract

The current approaches for datacenter power management (workload scheduling, CPU speed control, etc) focus primarily on maintaining the air temperature surrounding servers to be within the manufacturer specified constraint. This is problematic since several CPUs may still be violating the on-chip thermal constraint thereby leading to reliability loss. The primary objective of this work is to develop a unified approach for datacenter power optimization (by controlling the CPU speeds) which accounts for both the silicon level temperature of the VLSI components such as CPUs and the air temperature that directly impacts the reliability of other devices such as disks, and also the performance delivered. Our algorithm follows a two step approach: optimally solving a convex approximation that assigns continuous frequency values to all CPUs and a discretization step for legalization of the assigned frequencies. The experimental results indicate that our method guarantees both on-chip CPU and off-chip air temperature to be within temperature constraints. However, the traditional approach of constraining only air temperature will result in on-chip CPU temperature violation on about 40% of the CPUs, or 42% more power consumption to pull the CPU temperature back within constraint by increasing the HVAC cooling.

1 Introduction

Datacenters represent centralized facilities which have large number of high performance servers along with several petabytes of storage distributed across multiple racks. These facilities form the backbone of online services and serve millions of users everyday. For example, YouTube serves up to 100 million videos a day [1]; Facebook has 400 million active users and 3 billion photos uploaded each month [2]. These videos and images are stored and accessed from datacenters. Growing complexity and utilization of datacenters for performing computational and data accessing tasks has resulted in a significant increase in their power consumption levels. Modern datacenters roughly use 1.5% of the US electricity consumption according to recent EPA estimates [3]. With growing computing and storage capabilities, increasing connectivity, online services, advent of cloud computing, this energy footprint is slated to increase in the coming years. Energy consumption in datacenters comes from two broad sources 1) power dissipation in CPUs and the support circuitry (AC-DC converters, etc), and 2) the power dissipation in the HVAC system. Energy dissipated in electronic circuitry increases its operating temperature which inturn impacts the reliability and increases the failure rates of the devices. Therefore, manufacturers of datacenter servers and racks provide a maximum constraint on the air temperature surrounding the equipment. In order to maintain the air temperature, the HVAC system supplies cold air through vents and expends a significant amount of energy in doing so. Several approaches have been investigated that attempt to constraint the growing power demands in datacenters including load balancing, and the more recent CPU Vdd/speed control [4] [5] [6]. Other approaches that attempt to improve the efficiency of the HVAC system have also been investigated [7]. In this paper, we deal with the problem of CPU speed control in datacenters such that the overall power utilization is minimized while maintaining the performance and temperature. The current approaches for datacenter thermal management (workload scheduling, CPU speed control, etc) focus primarily on maintaining the air temperature surrounding servers to be within the manufacturer specified constraint. VLSI components such as CPUs, Memory etc have a maximum temperature constraint at silicon level as well. Constraining the air temperature certainly helps in ensuring the reliability of disks, AC-DC adapters, etc., but not of VLSI components such as CPUs. As we will show later, several CPUs might still violate the silicon level thermal constraint even though the air temperature is not very high. The primary objective of this work is to develop a unified approach for datacenter power optimization which accounts for both the silicon level temperature of the VLSI components such as CPUs and the air temperature that directly impacts the reliability of other devices such as disks. Thermal management in multi-core CPUs is an active topic of research where several models and optimization schemes have been developed for estimating and managing the chip level thermal profile. In this paper we attempt to develop a unified modeling and optimization approach for managing the overall datacenter power dissipation (by controlling the CPU speeds) while considering silicon level temperature constraint of CPUs, the traditional air temperature constraints and also the performance constraints. Our approach would result in a more reliable and lower power operation compared to traditional approaches for the same output performance.

Our algorithm follows a two step approach: first we approximate the CPU speed allocation problem as a continuous convex program which generates the frequency policy assuming it can be continuously controlled; then we discretize the frequency to discrete legal levels that the CPUs can run on. We account for both dynamic and leakage power and also model the leakage thermal interdependence. By exploiting the mathematical properties of convex programs, the convex approximation step can generate high quality solutions (which are then discretized) quickly. The convex program can have large number of unknowns since we are interested in simultaneously controlling both the chip level and air temperature while minimizing power. We describe ways of simplifying the problem without much impact on quality of solution. Experimental results show that power consumption estimated by our approximation is very close to the actual power consumption of the datacenter. Also, our method guarantees both the on-chip and air temperature to be within constraints, while simply ignoring leakage or constraining only on the air temperature will lead to overheating in about 40% to 60% CPUs. So in order to pull the on-

chip CPU temperature back within acceptable levels in this case, the datacenter needs to consume about 42% more power than our method. Our optimization framework implemented in MATLAB took about 4-5 minutes to execute for a 1000 server datacenter.

The paper is organized as follows. In section 2, we introduce the overall datacenter thermal/power model. In section 3, we explain the thermal/power model of multi-core CPUs and our optimization problem formulation. We develop a convex approximation approach to assign continuous frequency values to all CPUs in section 4. In section 5, we illustrate the frequency discretization approach while section 6 discusses some extension of our problem. The experimental result is given in section 7.

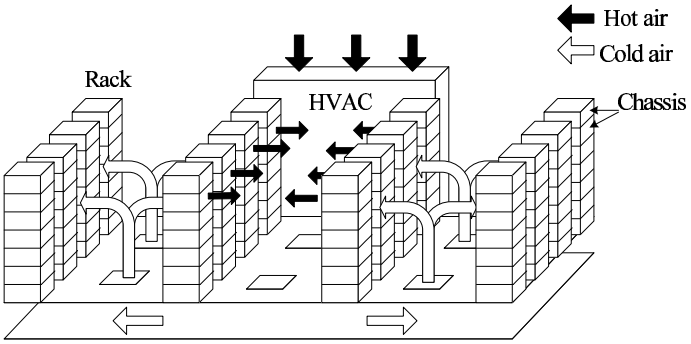


Figure 1: Datacenter model

2 Datacenter Power Management

Energy consumption in datacenters comes from two broad sources 1) power dissipation in CPUs and the support circuitry (DC converters etc.), and 2) the power dissipation in the HVAC system. Datacenters represent high performance servers packed together in hundreds of server racks. Scheduling of high performance tasks on servers results in excessive power dissipation in CPUs, memory chips, disks and also in the server support circuitry such as AC-DC converters, fans etc. All this dissipated energy results in higher operating temperature of the electronic circuitry. Higher operating temperature at silicon levels results in higher probability of error, reduced lifetime and reliability. Higher temperatures in datacenters also increases the chance of failure of other circuits such as fans, adapters etc. Therefore, manufacturers of datacenter serves and racks provide a maximum constraint on the air temperature surrounding the equipment. In order to maintain the air temperature, the HVAC system supplies cold air through vents and expends a significant amount of energy in doing so. Recent approaches try to perform task scheduling and/or CPU speed control such that a given amount of workload is completed without violating the air temperature constraint while minimizing the overall power utilized (electronic circuitry and HVAC system) [6]. Now we describe some basic equations that tie server power dissipation to the surrounding air temperature.

Datacenter racks incorporate several chassis which comprise of several server slots for housing servers (see figure 1 [6]). Servers comprise of several multi-core CPUs, RAM, disks etc, all of which dissipate power when used. Each chassis also has support circuitry such as power adapters etc. for maintaining the servers. If all the servers on a chassis are off, then this circuitry could be shut off, else it must be turned on. This dissipates around γ units of power ($\approx 820W$ as reported in [6] [8]). The server has components such as memory, disks etc which dissipate $\alpha \approx 60W - 120W$ of power [6] [8]. The CPUs in servers also dissipate power to the tune of 50-100W depending on their speed. The overall power dissipation in chassis n is given by equation 1 [6].

$$P_n = \gamma X_n + \alpha Y_n + \sum_{s=1}^{M} P_{n,s} \tag{1}$$

$$X_n = \begin{cases} 1, \text{if at least 1 server on chassis } n \text{ is on} \\ 0, \text{otherwise} \end{cases} \tag{2}$$

Here Y_n is the number of servers which are turned on in chassis n. Also, $P_{n,s}$ is the power dissipated in the multi-core CPU of the s-th server in the n-th chassis. Our model can trivially be extended to the case where servers have several multi-core CPUs. For the sake of simplicity in exposition we assume that the server has one multi-core CPU. The power consumed in severs and chassis is a strong function of the task scheduling and the CPU speed states. The basic datacenter organization is such that cool air coming from HVAC vents blows over the servers thereby heating up due to the server power dissipation. Let T_{in}^n be the temperature of the cool air entering the n-th chassis. The temperature of the air exiting the chassis T_{out}^n is given by [9] [10]:

$$P_n = K_n(T_{out}^n - T_{in}^n) \tag{3}$$

where K_n is a known constant based on the specific heat of air, rate of air flow etc. The input air temperature T_{in}^n is a function of the cool air temperature T_{sup} supplied by the HVAC vents. The close proximity of racks also results in intermixing of the hot air coming out of different chassis (see figure 2). This re-circulation causes the cool air into a chassis to intermix with the hot air from other chassis. The resulting input air temperature into a chassis n is given by equation 4.

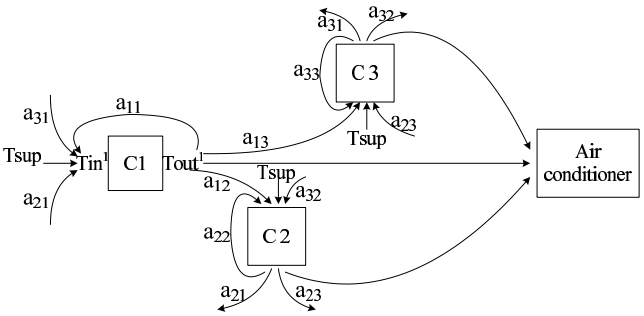


Figure 2: Air flow in datacenter

$$K_n T_{in}^n = \sum_{m=1}^{N} a_{mn} K_m T_{out}^m + (K_n - \sum_{m=1}^{N} a_{mn} K_m) T_{sup} \tag{4}$$

Here, T_{out}^m is the hot air coming from the m-th chassis and a_{mn} is the re-circulation factor or cross-interference coefficient between chassis m and n. These parameters depend on the design of the datacenter racks, rack placement, vent configuration etc and could be learnt using existing methodologies presented in [9]. Excessive power dissipation in chassis and also cross circulation of hot air results in an increase in datacenter air temperature, which needs to be maintained within manufacturer specified constraints. Therefore the HVAC system needs to reduce the supply air temperature T_{sup} thereby leading to increases energy consumption. The HVAC power is given by:

$$P_{AC} = \frac{\sum_{n=1}^{N} P_n}{COP} \tag{5}$$

Where COP is the coefficient of performance and $\sum_{n=1}^{N} P_n$ is the total power consumed by all the N chassis. COP is given by $COP = 0.0068T_{sup}^2 + 0.0008T_{sup} + 0.458$ [7]. The total power used by the datacenter is given by:

$$P_{total} = P_{AC} + \sum_{n=1}^{N} P_n = (1 + \frac{1}{COP}) \sum_{n=1}^{N} P_n \quad (6)$$

Existing approaches try to minimize this overall power such that $T_{out}^n \leq T_{cons}, \forall chasis\ n$ while maintaining acceptable performance levels. This can be achieved using a combination of task scheduling, CPU speed and T_{sup} control [6].

3 Datacenter Power Management: From Micro-scale to Mega-scale

In this paper, we deal with the problem of CPU speed control in datacenters such that the overall power utilization is minimized while maintaining the performance and temperature. We do not consider the problem of workload scheduling. A similar approach was investigated in [6]. The current approaches for datacenter thermal management (workload scheduling, CPU speed control etc) focus primarily on maintaining the air temperature surrounding chassis to be within the manufacturer specified constraint. VLSI components such as CPUs, Memory etc have a maximum temperature constraint at silicon level as well. Usually CPUs etc should not be heated beyond a certain temperature at silicon level for maintaining reliable operation [11] . Constraining the air temperature certainly helps in ensuring the reliability of disks, AC-DC adapters, etc., but not of VLSI components such as CPUs. Figure 3 illustrates the silicon level temperature of different server CPUs in a datacenter. This data was obtained by using the approach in [6] to assign CPU speed/Vdd such that overall power utilization is minimized while the air temperature is constrained to be ≤ 35℃. The total datacenter performance was also constrained to be higher than a certain value. The figure highlights the fact that constraining the air temperature does not necessarily ensure the CPU silicon temperature to be less than the manufacturer specified constraint. This would result in loss of reliability and higher device failure rates. This could certainly be fixed by reducing the supplied air temperature T_{sup} from the HVAC system. But this would be accompanied by an decrease in COP thereby resulting in an increase in the overall power dissipation.

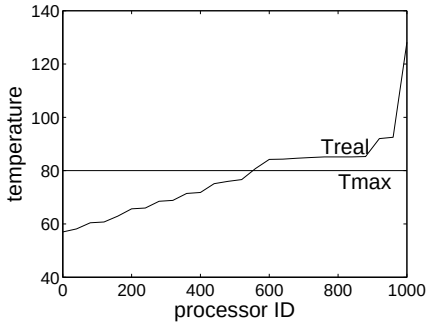


Figure 3: Silicon temperature of different Datacenter CPUs (Y-Axis oC)

The primary objective of this work is to develop a unified approach for datacenter power optimization which accounts for both the silicon level temperature of the VLSI components such as CPUs and the air temperature that directly impacts the reliability of other devices such as disks. Thermal management in multi-core CPUs is an active topic of research where several models and optimization schemes have been developed for estimating and managing the chip level thermal profile. In this paper we attempt to develop a unified modeling and optimization approach for managing the overall datacenter power dissipation while considering silicon level temperature constraint for CPUs, the traditional air temperature constraints and also the performance constraints. Our approach would result in a more reliable and lower power operation compared to traditional approaches for the same output performance. There are several challenges that we need to address in this regard.

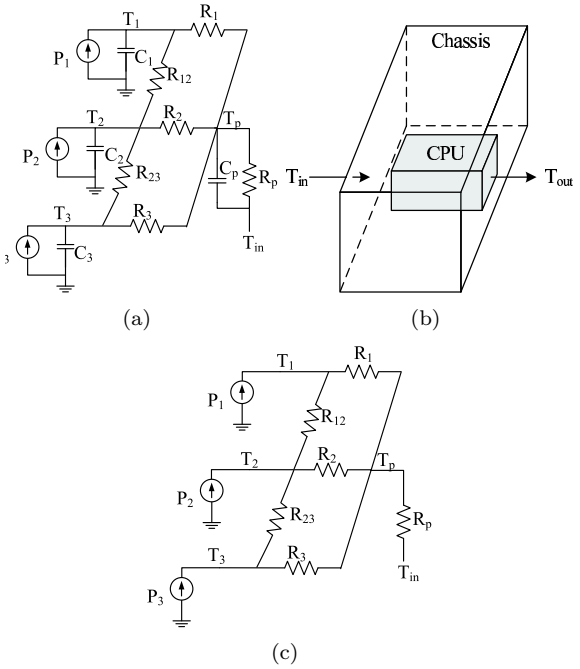


Figure 4: Multi-core CPU thermal model, (a)CPU dynamic thermal model, (b)impact of air temperature on CPU, (c) CPU steady state thermal model

1. Different time-scales: A unified approach needs to address the fact that on-chip silicon temperature roughly takes milliseconds to change while datacenter air temperature can take several minutes to change. A combined approach must address this lack of synchrony in the two events.
2. Impact of CPU leakage: Leakage power shows a strong thermal dependence. Increase in air temperature around CPUs will also indirectly increase the silicon temperature resulting in higher leakage. This would increase the overall power dissipation and impact both the silicon temperature and also the air temperature.
3. Complex optimization problem: CPUs demonstrate significant variation in temperature across the die. Constraining the maximum silicon temperature and also the air temperature would force us to formulate and solve a highly complex optimization problem with millions of unknown variables and therefore may not be a feasible option.

3.1 CPU Power-Thermal Model

Many researchers have developed thermal models that capture the on-chip temperature dynamics using a distributed RC circuit (see figure 4(a)) [11]. The individual current sources represent the

power dissipated in those areas and the voltage at the nodes represent temperature. This power is a function of the CPU operating frequency and also silicon temperature (due to leakage thermal interdependence). In this paper we assume that each core in a particular CPU runs at the same frequency although our methods are trivially extendable to the case where each CPU core has independent frequency as well. The thermal dynamics of the system shown in figure 4 is as follows:

$$\begin{aligned}\frac{dT_i}{dt} &= -\frac{T_i - T_p}{R_i C_i} - \sum_{\forall j \in NEI(i)} \frac{T_i - T_j}{R_{ij} C_i} + \frac{P_i}{C_i} \\ \frac{dT_p}{dt} &= -\frac{T_p - T_{in}}{R_p C_p} + \sum_{\forall i} \frac{T_i - T_p}{R_i C_p}\end{aligned} \tag{7}$$

Here T_{in} is the temperature of the air surrounding the CPU. This is the same with the input air temperature that the chassis intakes (see figure 4(b)). Here T_i is the temperature of the i-th node in the thermal model, and T_p is the package temperature. $NEI(i)$ refers to all the neighbor nodes of i-th node. Since the time scales of temperature change in CPUs and surrounding air is significantly different, we ignore the transient behavior and focus primarily on the steady state silicon temperature (see figure 4(c)). In the steady state, the silicon temperature T_i at all nodes i is given by (assuming T_{in} is a constant):

$$\begin{aligned}\frac{T_i - T_p}{R_i} + \sum_{\forall j \in NEI(i)} \frac{T_i - T_j}{R_{ij}} &= P_i \\ \frac{T_p - T_{in}}{R_p} - \sum_{\forall i} \frac{T_i - T_p}{R_i} &= 0\end{aligned} \tag{8}$$

By eliminating the variable T_p, we can represent the silicon temperature at all i (which represent different locations on chip) as follows:

$$w_{ii} T_i + \sum_{\forall j \in NEI(i)} w_{ij} T_j + w_i T_{in} = P_i \tag{9}$$

Here the parameters w_{ii}, w_i, w_{ij} can be derived from equations 8. Power dissipated at location i (P_i) depends on the average switching activity at location i which is a function of the operating frequency f. Leakage power which has strong thermal dependence is also a component of P_i.

$$P_i = P^i_{leak} + P^i_{dynamic} = b_i T_i^2 e^{-\frac{c_i}{T_i}} + d_i + \beta_i f \tag{10}$$

Here b_i, c_i are device dependent constants which control the leakage thermal interdependence [12], β_i is the amount of capacitance that we switch at location i and d_i is a constant that depends on other circuit parameters. This model captures the steady state temperature profile of CPUs at silicon level as a function of the power dissipation profile and also the ambient air temperature T_{in}. As indicated in equation 4, the temperature around the CPU T_{in} would be a function of both T_{sup} and T_{out} of other chassis.

3.2 Optimization Formulation

In this paper, we develop formulations that synthesize the optimal frequency policy for all CPUs in the servers such that the overall power utilization is minimized and 1) the silicon temperature at all CPUs is less than a constraint T_{max} 2) the air temperature T^n_{out} at all chassis n is less than T^{chasis}_{max} and 3) the total frequency of all CPUs is greater than a specified constraint. For the sake of exposition, we will assume that the HVAC supplied temperature T_{sup} is given to us. The formulation is easily extendible even if T_{sup} is a controllable parameter. We shall also assume that the air temperature inside the chassis n is T^n_{in}. Therefore the ambient temperature for all server CPU inside chassis n is T^n_{in}.

Objective:

Our objective is to minimize the total power dissipated in the datacenter.

$$Minimize \quad (1 + \frac{1}{COP} \sum_{n=1}^{N} P_n) \tag{11}$$

Since T_{sup} is assumed to be known, COP is a known constant (see discussion in section 2).

Constraints:

We assume that all CPUs can be in discrete frequency state belonging to the set $0, f_1, f_2 ..... f_K$. The problem constraints can be written as follows.

$$\begin{aligned}
&1.\ \sum_{n=1}^{N} \sum_{s=1}^{M} f_{n,s} \geq F \\
&2.\ T_{n,s,i} \leq T_{max}, \forall n, s, i \\
&3.\ w_{ii} T_{n,s,i} + \sum_{\forall j \in NEI(i)} w_{ij} T_{n,s,j} + w_i T^n_{in} = P_{n,s,i}, \forall n, s, i \\
&4.\ P_{n,s,i} = b_{n,s,i} T^2_{n,s,i} e^{-\frac{c_{n,s,i}}{T_{n,s,i}}} + d_{n,s,i} + \beta_{n,s,i} f_{n,s}, \forall n, s, i \\
&5.\ P_n = \gamma X_n + \alpha Y_n + \sum_{s=1}^{M} \sum_{i=1}^{L} P_{n,s,i}, \forall n \\
&6.\ K_n (T^n_{out} - T^n_{in}) = P_n, \forall n \\
&7.\ K_n T^n_{in} = \sum_{m=1}^{N} a_{mn} K_m T^m_{out} + (K_n - \sum_{m=1}^{N} a_{mn} K_m) T_{sup}, \forall n \\
&8.\ T^n_{out} \leq T^{chasis}_{max}, \forall n \\
&9.\ f_{n,s} \in \{f_0, f_1, f_2 ..... f_K\}, \forall n, s
\end{aligned} \tag{12}$$

There are a total of M servers per chassis and a total of N chassis in the datacenter. Here $f_{n,s}$ is the frequency of the s-th server CPU on the n-th chassis. $T_{n,s,i}$ is the temperature of the i-th on-chip node (see figure 4(c)) on the s-th server CPU of the n-th chassis. $P_{n,s,i}$ is the power (leakage and dynamic) of the i-th on-chip node of the s-th server CPU of the n-th chassis. T^n_{in} and T^n_{out} are the intake and exhaust air temperatures for the n-th chassis. The first constraint ensures that the total frequency delivered by the datacenter is at least greater than F. The second constraint guarantees the on-chip silicon CPU temperature to be $\leq T_{max}$. As illustrated in equations 9 and 10 the third and fourth constraints establish the interdependence between temperature and power of the i-th on chip location of the s-th server CPU on the n-th chassis. Similar to equation 1, the fifth constraint specified the total power dissipated in the n-th chassis. In this constraint, X_n represents the power consumption overhead of chassis n when at least one server of this chassis is on, while Y_n is the number of chassis that is turned on in chassis n (see section 2). The sixth constraint is similar to equation 3 and establishes the relationship between the intake air temperature and the exhaust output air temperature for the n-th chassis. Here we have assumed that the ambient temperature inside the n-th chassis for all the server CPUs is T^n_{in}. The seventh constraint establishes the relationship between the input air temperature and the out air temperature of all the chassis and T_{sup} (see equation 4 for details). The eighth constraint limits the output air temperature at all chassis to be within T^{chasis}_{max}.

This formulation is highly nonlinear and the integer constraints significantly increase the complexity. Now we present a two step algorithmic approach for finding the best frequency policy. First we relax the integer constraint imposed on the CPU frequency to

find a reasonable continuous frequency policy. This is followed by a discretization step that legalizes the frequency policy.

4 Continuous Convex Approximation

The problem formulation in equations 11 and 12 is highly complex and cannot be solved optimally in polynomial time. Even if the the frequencies could be controlled continuously in the range $0 \leq f \leq f_{max}$, the non-linear leakage-thermal interdependence leads to a set of nonlinear constraints. Also variables X_n and Y_n must be integer values even if the frequencies are continuous (see section 2 for a discussion on the nature of X_n and Y_n). Now we develop a continuous approximation of the original formulation that addresses these challenges systematically. We begin by assuming that all the CPU frequencies $f_{n,s}$ are continuous values that lie in the range $0...f_{max}$. Now we make the following transformation.

$$\begin{aligned} &f_{n,s} = e^{log(f_{max}+1)\eta_{n,s}} - 1 \\ &\text{if } 0 \leq f_{n,s} \leq f_{max}, \text{then } 0 \leq \eta_{n,s} \leq 1 \end{aligned} \tag{13}$$

Nonlinear Leakage Power-Thermal Interdependence:

Constraints 3 and 4 in equation 12 indicate the power dissipated $P_{n,s,i}$ in the i-th on-chip node of the s-th server CPU of the n-th chassis is a function of the frequency $f_{n,s}$ and temperature $T_{n,s,i}$. Combining constraints 3 and 4 and expressing $f_{n,s}$ as $\eta_{n,s}$ gives us the following equation.

$$\begin{aligned} b_{n,s,i}T_{n,s,i}^2 e^{-\frac{c_{n,s,i}}{T_{n,s,i}}} + d_{n,s,i} + \beta_{n,s,i}(e^{log(f_{max}+1)\eta_{n,s}} - 1) \\ -w_{ii}T_{n,s,i} - \sum_{\forall j \in NEI(i)} w_{ij}T_{n,s,j} - w_i T_{in}^n = 0 \end{aligned} \tag{14}$$

Theorem 1: The left had side of equation 14 is a convex function in $T_{n,s,i}$, $T_{n,s,j}$, $\eta_{n,s}$ and T_{in}^n.

Proof: $b_{n,s,i}T_{n,s,i}^2 e^{-\frac{c_{n,s,i}}{T_{n,s,i}}}$, $\beta_{n,s,i}(e^{log(f_{max}+1)\eta_{n,s}} - 1)$, $-w_{ii}T_{n,s,i} - \sum_{\forall j \in NEI(i)} w_{ij}T_{n,s,j} - w_i T_{in}^n$ can be shown to be convex functions. A positive linear combination of convex functions is a convex function. Hence proved.

We shall exploit this convexity property later.

Accounting for X_n and Y_n:

Even if we approximate the frequency to be a continuous variable, parameters X_n and Y_n must be discrete. As discussed earlier $X_n = 1$ if even one server on chassis n is turned on (see equation 1,2). Also Y_n is the total number of servers that are turned on, regardless of the frequency. Let us define a new parameter $Y_{n,s}$ where $Y_{n,s} = 1$ indicates that the s-th server on the n-th chassis is turned on (that is, $Y_{n,s} = 1$ if $f_{n,s}$ or $\eta_{n,s} > 0$) and 0 otherwise. The following is the relationship between $Y_{n,s}$ and X_n, Y_n:

$$\begin{aligned} Y_n &= \sum_{s=1}^{M} Y_{n,s} \\ X_n &= \max_{\forall s} Y_{n,s} \end{aligned} \tag{15}$$

Now we develop a way of approximating the function $Y_{n,s}$ (which is a function of $f_{n,s}$ or $\eta_{n,s}$) and thereby develop a way of approximating the discrete nature of variables X_n and Y_n.

Figure 5 indicates the variable $Y_{n,s}$ as a function of $f_{n,s}$. It also indicates our approximation $Y'_{n,s}$ of $Y_{n,s}$.

$$Y'_{n,s} = \frac{log(f_{n,s}+1)}{log(f_{max}+1)} = \eta_{n,s} \tag{16}$$

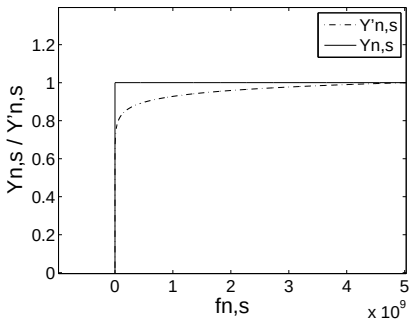


Figure 5: $Y_{n,s}$ and its approximation $Y'_{n,s}$

$$Y'_{n,s} = \begin{cases} 1, \text{when } f_{n,s} = f_{max} \\ 0, \text{when } f_{n,s} = 0 \end{cases} \tag{17}$$

Basically $Y'_{n,s}$ is a concave function that approximates the actual function as much as possible. $Y'_{n,s}$ is represented using the function in equation 16. It can clearly be seen that $Y'_{n,s} = \eta_{n,s}$. This transformation is very important since $Y'_{n,s}$ becomes a linear function of $\eta_{n,s}$. Now Y_n and X_n can be approximated as follows.

$$\begin{aligned} X_n &\approx \max_{\forall s} Y'_{n,s} = \max_{\forall s} \eta_{n,s} \\ Y_n &\approx \sum_{s=1}^{M} Y'_{n,s} = \sum_{s=1}^{M} \eta_{n,s} \end{aligned} \tag{18}$$

Theorem 2: Y_n is approximated as a linear function of $\eta_{n,s}$ and X_n approximated as a convex function of $\eta_{n,s}$

Proof: Omitted for brevity.

4.1 Overall Continuous Convex Formulation

The overall formulation that approximates the original problem is as follows. Firstly, we make the observation that the total power dissipated $(1 + \frac{1}{COP}) \sum_{n=1}^{N} P_n) = (1 + \frac{1}{COP}) \sum_{n=1}^{N} K_n(T_{out}^n - T_{in}^n)$ since $P_n = K_n(T_{out}^n - T_{in}^n)$ (see equation 3). Hence the objective becomes.

$$Minimize \quad (1 + \frac{1}{COP}) \sum_{n=1}^{N} K_n(T_{out}^n - T_{in}^n) \tag{19}$$

The new set of constraints obtained by transforming $f_{n,s}$ to $\eta_{n,s}$ and incorporating the approximations for X_n and Y_n are as follows:

$$
\begin{aligned}
&1.\ \sum_{n=1}^{N}\sum_{s=1}^{M}(e^{log(f_{max}+1)\eta_{n,s}}-1) \geq F \\
&2.\ T_{n,s,i} \leq T_{max}, \forall n,s,i \\
&3.\ b_{n,s,i}T_{n,s,i}^2 e^{-\frac{c_{n,s,i}}{T_{n,s,i}}} + d_{n,s,i} + \beta_{n,s,i}(e^{log(f_{max}+1)\eta_{n,s}}-1) \\
&\quad - w_{ii}T_{n,s,i} - \sum_{\forall j \in NEI(i)} w_{ij}T_{n,s,j} - w_i T_{in}^n = 0, \forall n,s,i \\
&4.\ \gamma \max_{\forall s}\eta_{n,s} + \alpha\sum_{s=1}^{M}\eta_{n,s} + \sum_{s=1}^{M}\sum_{i=1}^{L}(w_{ii}T_{n,s,i} + \\
&\quad \sum_{\forall j \in NEI(i)} w_{ij}T_{n,s,j} + w_i T_{in}^n) - K_n(T_{out}^n - T_{in}^n) = 0, \forall n \\
&5.\ K_n T_{in}^n = \sum_{m=1}^{N} a_{mn}K_m T_{out}^m + (K_n - \sum_{m=1}^{N} a_{mn}K_m)T_{sup}, \forall n \\
&6.\ T_{out}^n \leq T_{max}^{chasis}, \forall n \\
&7.\ 0 \leq \eta_{n,s} \leq 1, \forall n,s
\end{aligned}
\tag{20}
$$

Note that constraint 3 above is a combination of constraint 3 and 4 in equation 12. Also constraint 4 above is the same as the combined set of constraints 4,5,6 in equation 12. Now we make the following transformations. Consider constraint 1. Clearly it is not a convex constraint. But we make the following transformation to fix it.

$$
-(\sum_{n=1}^{N}\sum_{s=1}^{M} e^{log(f_{max}+1)\eta_{n,s}}) \leq -(F+MN) \tag{21}
$$

Since $\sum_{n=1}^{N}\sum_{s=1}^{M} e^{log(f_{max}+1)\eta_{n,s}}$ is a monotonically increasing function in $\eta_{n,s}$, $-(\sum_{n=1}^{N}\sum_{s=1}^{M} e^{log(f_{max}+1)\eta_{n,s}})$ is a monotonically decreasing function in $\eta_{n,s}$. Hence the constraint in equation 21, which is equivalent to the first constraint in equation 20, becomes *quasiconvex* [13]. Quasiconvex constraints can be treated just as convex constraints for all practical purposes [13].

Now consider constraints 3 and 4 in equation 20. Instead of representing them as $= 0$, we represent them as follows.

$$
\begin{aligned}
&3.\ b_{n,s,i}T_{n,s,i}^2 e^{-\frac{c_{n,s,i}}{T_{n,s,i}}} + d_{n,s,i} + \beta_{n,s,i}(e^{log(f_{max}+1)\eta_{n,s}}-1) \\
&\quad - w_{ii}T_{n,s,i} - \sum_{\forall j \in NEI(i)} w_{ij}T_{n,s,j} - w_i T_{in}^n \leq 0, \forall n,s,i \\
&4.\ \gamma \max_{\forall s}\eta_{n,s} + \alpha\sum_{s=1}^{M}\eta_{n,s} + \sum_{s=1}^{M}\sum_{i=1}^{L}(w_{ii}T_{n,s,i} \\
&\quad + \sum_{\forall j \in NEI(i)} w_{ij}T_{n,s,j} + w_i T_{in}^n) - K_n(T_{out}^n - T_{in}^n) \leq 0, \forall n
\end{aligned}
\tag{22}
$$

Using theorem 1 and 2, we can easily see that these represent a set of convex constraints.

Hence we can represent the approximate optimization problem highlighted in equations 19 and 20 as a convex optimization problem. This is because, the modifications in constraints 1, 3 and 4 result in convex constraints. All the other constraints are linear and the objective is linear as well. Hence it could be solved optimally in polynomial time. One might argue that the optimal solution of such a formulation might be problematic due to the inequalities in equation 22. These equations establish the interdependence between CPU silicon temperature, frequency and the input air temperature. Therefore they must really be represented as equalities to 0 rather than inequalities. The following theorem fixes this problem.

Theorem 3: In the optimal solution of the convex formulation described above, the inequalities of equation 22 become equalities to 0.

Proof: Omitted for brevity.

This completes our continuous formulation that assigns the frequencies such that performance and thermal constraints are satisfied and overall power is minimized.

4.2 Computational Complexity

Even though, convex optimization problems can be optimally solved in polynomial time, the scale of the problem in our case is very large. A typical datacenter can have hundreds of racks comprising of thousands of servers. Accounting for CPU silicon level temperature constraint would significantly increase the unknown variables and therefore could make solving the convex optimization formulation practically in-feasible. In this context there are several simplifications we can do to reduce the size of the optimization problem. The thermal inter-coupling between racks would only exist, in general, between neighboring racks. Hence the constraints that represent the interdependence between T_in and T_out of racks could be made more sparse thereby leading to simplification of the optimization process.

Also, the quality of solution and runtime is the function of how complex the on-chip thermal model is. In many cases we might be interested in simply constraining the package temperature of CPUs rather than the temperature at all internal regions of interest on silicon. This would simplify the CPU thermal model to a simple RC circuit. This would add only a few extra unknown variables over the formulation in [6] which could be handled easily by the convex optimization tool. In many cases we must constrain the silicon temperature at different points of interest on chip as well. In such scenarios we could have a simpler RC thermal model rather than the complex model in [11]. For example, consider the thermal model in figure 4(c). Here each node represents an on-chip CPU core in a multi-core chip. A homogeneous multi-core design would imply the individual RC parameters for each core to be the same. Hence the on-chip temperature of each node i (on-chip core) could be assumed to be the same as well. Therefore for each CPU we have only one temperature variable. This would significantly reduce the overall complexity. Note that similar RC thermal models for multi-cores were presented by [14]. Assuming homogeneous CPUs in all servers further reduces the overall problem complexity.

These techniques help in solving the complex optimization problem that combines the chip level and datacenter level abstraction in a unified framework quickly and efficiently. Although such approximations would result in reduction in accuracy, the level of granularity in controlling the on-chip temperature does not need to be very high since we are considering the problem at the level of datacenters. We implemented many of these techniques for improving the runtime. But, in this paper we do not investigate the full scope of applying these techniques for runtime improvement.

5 Frequency Discretization

In general, most CPUs are constrained to operate on a pre-decided set of discrete frequencies. Hence, the frequency should be selected from some pre-defined set of discrete levels. So we wish to discretize the continuous frequency into discrete levels. The discretization is basically approximating the frequency to the lowest discrete level that is greater than the original continuous value. This approximation ensures the performance, so that the total frequency delivered by the datacenter is ensured to be greater than

the system requirement. However, this discretization may result in violation of the maximum on-chip silicon CPU temperature or the chassis output air temperature constraint. If this occurs, we reduce the T_{sup} to pull the CPU and air temperature back within the constraints at the expense of increasing HVAC power consumption.

6 Extensions

Several extensions on our basic formulation presented above are possible. Firstly, combining the task scheduling and CPU speed control techniques would improve the quality of solution. Development of such a combined optimization technique is out of scope of this work. Also, our optimization problem can be easily extended to other specifications. The formulation in [6] performs CPU speed assignment such they follow a specific service level agreement. Specifically, the following constraints are imposed on the total CPU frequencies.

$$\sum_{n=1}^{N}\sum_{s=1}^{M} g_{n,s}^{x} \geq Q_x, x = 1...X \tag{23}$$

where

$$g_{n,s}^{x} = \begin{cases} 1, \text{if s-th server on n-th chassis runs on frequency } f \geq f_x \\ 0, \text{otherwise} \end{cases} \tag{24}$$

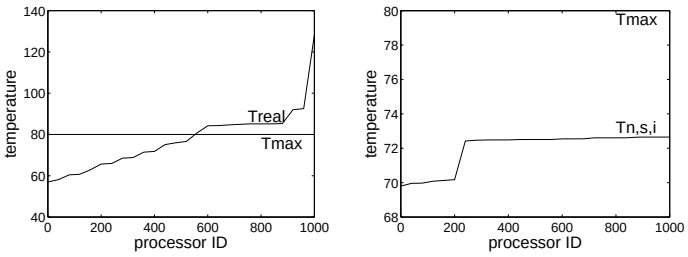
Basically, the frequencies need to be assigned such that at least Q_x CPUs should have frequency greater than or equal to a frequency level f_x. For example one might want 60% of the CPUs to run at f_{max} and 30% at $f_{max}/2$. This integer performance constraint $g_{n,s}^{x}$ can be approximated by continuous convex constraint using a method similar to approximating X_n and Y_n (as described in section 4). We do not describe further details of such techniques for the sake of brevity.

7 Experimental Results

In the experiment, we use a small scale datacenter similar as in [6] [8]. The datacenter has two rows, each row consists of 5 racks. Each rack consists of 5 chassis and each chassis contains 20 servers. Each server on the chassis is a dual-core processor. Therefore, there are totally 1000 dual-core server CPUs in the datacenter. We assume the two cores on each CPU are homogenous, so they have the same temperature profile. The chassis power overhead γ is 820W and the non-core power overhead of server α is 60W [6]. The discrete frequency set is {0, 2GHz, 3GHz, 4GHz, 5GHz}.

7.1 Comparison of our method with purely air temperature constraint

We firstly compare our method with the method that only imposes constraint on the output air temperature [6] (we call it '*off-chip*' method). In our experiment, the supply temperature $T_{sup} = 10$℃, and the chassis output air temperature constraint $T_{max}^{chassis} = 35$℃. The total frequency constraint is $F = 5 \times 10^{12} Hz$ (note we have 1000 CPU servers and 2000 CPU cores, so this is equivalent to an average frequency constraint of $2.5GHz$). Figure 6(a) shows the on-chip temperature distribution on all the server CPU cores in the datacenter (T_{real}) achieved by this method. Since we assume each CPU is a homogenous dual-core processor, the temperature on the two cores of each CPU are the same and therefore, we just plot the temperature of one core for each dual-core processor. Assuming the maximum on-chip silicon temperature constraint is $T_{max} = 80$℃, as we can see, the on-chip silicon temperature of more than 40% of the CPUs violates the maximum temperature constraint. Some cores even heats up to about 130℃.



(a) (b)

Figure 6: (a)On-chip temperature profile achieved by constraining only on output air temperature, (b)Temperature profile achieved by our method

However, as we can see from figure 6(b), using our method, the on-chip silicon temperature $T_{n,s,i}$ will stay within the temperature constraint, since we impose constraints on both the on-chip silicon temperature and the air temperature.

On the other hand, in *off-chip* method, one can also try to pull the on-chip temperature below maximum temperature constraint by reducing the HVAC supply temperature T_{sup}. However, this will result in increase of HVAC power consumption and therefore, increase the total power consumption of the datacenter. Table 1 compares the power consumption achieved by our method and *off-chip* method when setting $T_{sup} = 10$℃in the optimization. P_{old} is the total power consumption achieved without trying to fixing the on-chip temperature, while P_{new} is the total power consumption achieved after fixing the on-chip temperature by reducing T_{sup}. As we can see, since our method does not lead to on-chip temperature constraint violation, we don't need to reduce T_{sup}. However, for the *off-chip* method, T_{sup} is reduced to pull the on-chip temperature down and results in about 42% power consumption increase.

Table 1: Total power consumption of our method and *off-chip* method

Method	P_{old}(W)	P_{new}(W)
Our	5.5964×10^5	5.5964×10^5
Off-chip	5.6435×10^5	8.0434×10^5

7.2 Comparison of our method with ignoring leakage method

We then look at the temperature profile achieved by the method ignoring leakage power consumption (called *no-leak*). We use the same T_{sup}, T_{max} and $T_{max}^{chassis}$ settings with section 7.1. When the total frequency constraint is $F = 5 \times 10^{12} Hz$, we calculate the optimal frequency scheme using the method ignoring leakage power, and then estimate the actual on-chip silicon temperature profile considering leakage. The resulting temperature profile is shown in figure 7. As we can see, the frequency assignment achieved by *no-leak* method will result in violation of maximum temperature constraint in about 60% of the CPUs. The on-chip temperature

of some cores will reach as high as 105℃. Compared with *off-chip* method, although more CPU cores violate the on-chip temperature constraints, the degree of violation is smaller.

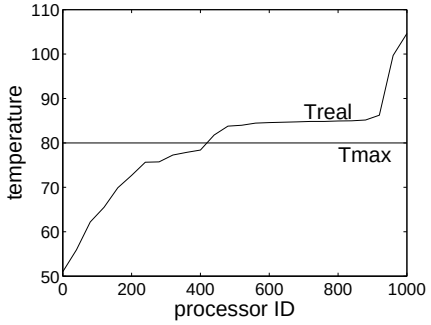
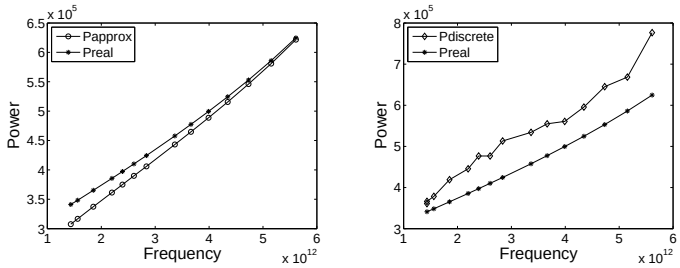


Figure 7: Real temperature profile achieved by *no-leak* method

7.3 Comparison of our approximated power with real power consumption

In our model, we approximate the integer constraints X_n and Y_n with continuous function as shown in figure 5 and equations 16,18. We then test the performance of our approximation by comparing the power consumption estimated by our method with the real power consumption where we use the actual formula for X_n and Y_n . We calculate the optimal frequency assignment for the datacenter that minimizes the total datacenter power consumption by our method, and then compare the power consumption approximated by our method with the real power consumption of the datacenter under this frequency assignment. Figure 8(a) shows the power consumption approximated by our method and the actual power consumption for different total frequency constraints. In this figure, P_{approx} is the datacenter power consumption approximated by our method, P_{real} is the actual power consumption under this frequency scheme. As we can see from this figure, the approximated power consumption is very close to the real power consumption, and only underestimates the total power consumption by 4% on average. Also, when the system performance constraint (total frequency constraint) increases, the approximation works better and when the total frequency is about $5.6 \times 10^{12} Hz$ (that is, the average frequency of each CPU is $2.8GHz$), our approximation is only 0.5% lower.



(a) (b)

Figure 8: (a)Comparison of approximated and real power consumption, (b)Comparison of power consumption after frequency discretization and power consumption under continuous frequency assignment

In the discretization, we round the frequency up to the nearest discrete frequency value greater than the continuous value in order to guarantee the performance. Figure 8(b) shows the power consumption after frequency discretization compared to the power consumption of continuous frequency scheme for different total frequency constraints. The power consumption is about 12% more after discretization on average.

Our optimization framework implemented in MATLAB took about 4.8 minutes to execute for a 1000 dual-core server datacenter and 21.6 minutes for a 2000 dual-core server datacenter.

8 Conclusion

In this paper, we develop a unified approach for datacenter power optimization which accounts for the silicon level temperature of the VLSI components, the air temperature, the performance delivered, and also the leakage thermal interdependence. We use a two step approach to solve the problem by: 1) optimally solving a convex approximation that assigns continuous frequency values to all CPUs and 2) discretizing the assigned frequencies. By exploiting the mathematical properties of convex programs, the convex approximation step generates high quality solutions quickly.

Acknowledgement

This research work was partly supported by NSF grant CCF 0937865.

References

[1] "Youtube serves up 100 million videos a day online," *USA Today*, 2006.

[2] "Facebook statistics," *http://www.facebook.com.*

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

[4] V. Cardellini, M. Colajanni, and P. S. Yu, "Dynamic load balancing on web-server systems," *IEEE Internet Computing Magazine*, vol. 3, pp. 28–39, 1999.

[5] D. M. Dias, W. Kish, R. Mukherjee, and R. Tewari, "A scalable and highly available web server," in *Proceeding of IEEE Computer Society International Conference*, pp. 85–92, 1996.

[6] E. Pakbaznia and M. Pedram, "Minimizing data center cooling and server power costs," in *Proceedings of the 2003 International Symposium on Low Power Electronics and Design (ISLPED'09)*, pp. 145–150, 2009.

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

[8] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Energy-efficient, thermal-aware task scheduling for homogeneous, high performance computing data centers: A cyber-physical approach," in *IEEE Transactions on Parallel and Distributed Systems*, pp. 1458–1472, 2008.

[9] Q. Tang, T. Mukherjee, S. K. S. Gupta, and P. Cayton, "Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters," in *International Conference on Intelligent Sensing and Information (ICISIP2006)*, pp. 203–208, 2006.

[10] Q. Tang, S. K. S. Gupta, D. Stanzione, and P. Cayton, "Thermal-aware task scheduling to minimize energy usage of

blade server based datacenters," in *Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*, pp. 195–202, 2006.

[11] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. on Architecture and Code Optimization*, vol. 1, pp. 94–125, 3.

[12] L. He, W. Liao, and M. R. Stan, "System level leakage reduction considering the interdependence of temperature and leakage," in *Design Automation Conference (DAC'04)*.

[13] S. Boyd and L. Vandenberghe, "Convex optimization," *Cambridge University Press, New York, NY*, 2004.

[14] R. Rao, S. Vrudhula, and N. Chang, "An optimal analytical solution for processor speed control with thermal constraints," in *Proc. of Intl. Symp. on Low Power Electronics and Design (ISLPED'06)*.