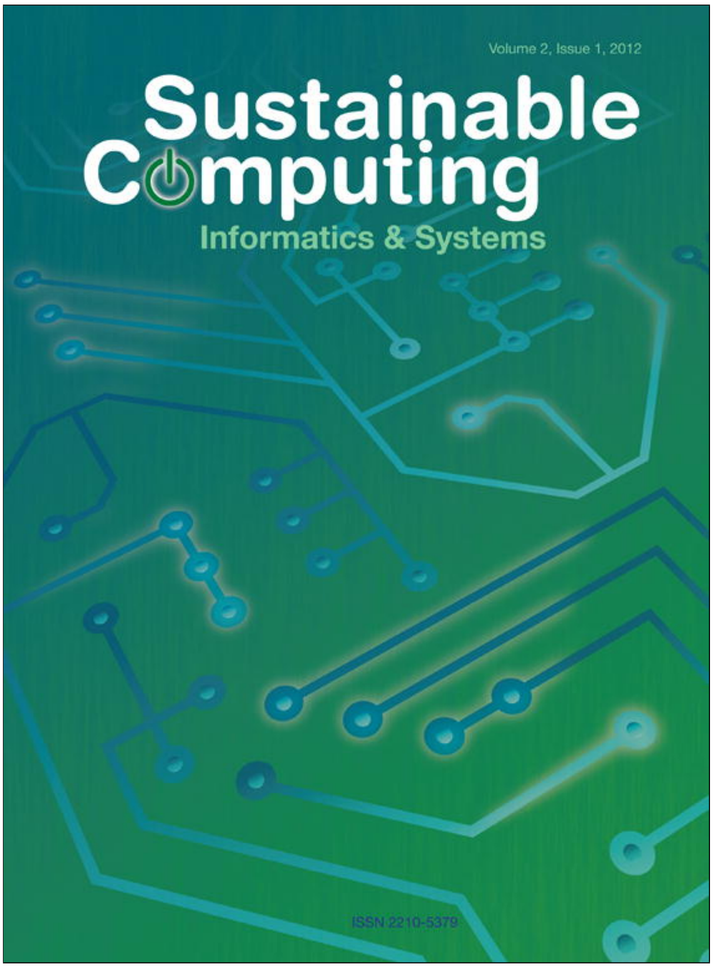
Volume 2, Issue 1, 2012
Sustainable Computing
Informatics & Systems
ISSN 2210-5379

Sustainable Computing: Informatics and Systems 2 (2012) 13–22

Contents lists available at SciVerse ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

Optimal power allocation among multiple heterogeneous servers in a data center

Keqin Li*

Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Keywords:
Average task response time
Data center
Heterogeneous server
Optimal power allocation
Power consumption

ABSTRACT

In a data center for cloud computing, there are typically multiple heterogeneous servers which provide services in different application domains. For such heterogeneous servers in a data center with different configurations for diversified applications and certain available power, there is a problem of allocating the power to the servers, such that the overall quality of service of the servers in the data center is optimized. We address power constrained performance optimization in a data center with multiple heterogeneous servers. We consider the problem of optimal power allocation among multiple heterogeneous servers, i.e., minimizing the average task response time of multiple heterogeneous computer systems with energy constraint. Each server is treated as a queueing system and the average task response time in a data center with multiple servers is formulated as a function of power allocations to the servers. The average task response time is minimized subjected to the constraint that the total effective power consumption of all the servers does not exceed a given power limit. We develop an algorithm to find the optimal solution and demonstrate numerical data. We also develop several closed-form heuristic solutions and show that they are very close to the optimal solution. Our approach provides an analytical way of studying the power-performance tradeoff at the data center level.

1. Introduction

Data center energy consumption in the US has doubled every five years, reaching about 110 billion kilowatt-hours per year by 2011 and representing an annual electricity cost of 7.4 billion US dollars. The peak load on the power grid from servers of data centers will be 12 GW by 2011, equivalent to the output of 25 baseload power plants [42]. A data center can consume up to 100 times more energy than a standard office building. Data centers consume large amount of natural resources. Only 33% of the original source energy can be transformed to electricity and delivered to data centers for consumption. More than half of the electrical power is used by cooling equipment and for power conversion and distribution. Eventually, less than 15% of the original source energy is actually used by information technology equipment for useful computation and communication within a data center. In addition to consumption of natural resources, data centers in many areas around the world are powered by coal-fired or natural gas electrical generation plants, creating tremendous amounts of CO_2 emissions as a bi-product of their power consumption.

Cloud computing provides a fundamentally new way of delivering computing and information technology services and has been rapidly and widely considered and accepted as a promising computing paradigm [38]. In a data center for cloud computing, there are typically multiple heterogeneous servers which provide services in different application domains. Due to different characteristics of various applications, a server is specifically configured in terms of computing power, memory capacity, communication bandwidth, and so on, to fit the requirements of a specific application domain. Each server accepts service requests from a particular application domain with unique workload characteristics such as the task arrival rate and the average task execution requirement. For such heterogeneous servers with different configurations for diversified applications, load distribution among servers seems inadequate, i.e., each service request must be submitted to a designated server. Another case is the type of dedicated servers, which provide dedicated hosting service or managed hosting service, i.e., a type of Internet hosting in which a client leases an entire server not shared with anyone [2]. The average task response time of a server can be improved by increasing the processor computing power of the server. However, increasing the processor speed of a server implies more power consumption of the server. Hence, given certain available power, there is a problem of allocating the power to the servers in a data center, such that the overall quality of service of the servers in the data center is optimized.

In this paper, we deal with power constrained performance optimization in a data center with multiple heterogeneous servers. We consider the problem of optimal power allocation among multiple

* Tel.: +1 845 257 3534; fax: +1 845 257 3996.

2210-5379/$ – see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.suscom.2011.11.002

heterogeneous servers, i.e., minimizing the average task response time of multiple heterogeneous computer systems with energy constraint. Notice that from users' point of view, the average task response time of all servers is an important measure of quality of service in a data center. Each server is treated as a queueing system and the average task response time in a data center with multiple servers is formulated as a function of power allocations to the servers. The average task response time is minimized subjected to the constraint that the total effective power consumption of all the servers does not exceed a given power limit. Such a minimization problem is solved by finding an optimal power allocation to the servers, since a power allocation determines both average task response time and total effective power consumption. We develop an algorithm to find the optimal solution and demonstrate numerical data. We also develop several closed-form heuristic solutions and show that they are very close to the optimal solution. Our approach provides an analytical way of studying the power-performance tradeoff at the data center level. To the best of the author's knowledge, such combined investigation of data center performance optimization and energy efficiency has not been studied before as a multivariable optimization problem in the literature.

The rest of the paper is organized as follows. In Section 2, we mention related research. In Section 3, we present the power consumption model used in this paper. In Section 4, we describe a queueing model for each server so that the average task response time can be characterized analytically. In Section 5, we formally define our optimal power allocation problem. In Section 6, we develop an algorithm to solve our optimization problem. In Section 7, we demonstrate numerical data. In Section 8, we develop several closed-form heuristic solutions and show their quality. In Section 9, we conclude the paper.

2. Related research

According to Moore's law, power consumption in computer systems has increased at an exponential speed for decades. Power density in high-performance computer systems will soon reach that of a nuclear reactor [43]. Such increased energy consumption causes severe economic, ecological, environmental, and technical problems [12–14,39]. Power conservation is critical in many computation and communication environments and has attracted extensive research activities. Reducing processor energy consumption has been an important and pressing research issue in recent years. There has been increasing interest and importance in developing high-performance and energy-efficient computing systems and data centers. Significant research and development efforts have been devoted to finding power and performance management methods, and an explosively growing body of literature has been developed for energy-efficient computing and communication. The reader is referred to [3,7,41,43] for comprehensive surveys.

Power consumption in computing systems can be reduced by thermal-aware hardware and software design at various levels [40]. Among the numerous hardware and software techniques, methods, and paradigms ever developed for reducing energy consumption, dynamic power management at the operating system level is one of the most effective and efficient ways of managing the power-performance tradeoff. Such software techniques for power reduction are supported by a mechanism called *dynamic voltage scaling* (equivalently, dynamic frequency scaling, dynamic speed scaling, dynamic power scaling), which are based on supply voltage and clock frequency and processor speed and power consumption adjustment schemes implemented while tasks are running. A power-aware scheduling or control algorithm can change supply voltage and clock frequency and processor speed and power consumption at appropriate times to optimize a combined consideration of power reduction and performance optimization. Such management of power and performance can be carried out at different levels, i.e., task level, system and server level, server cluster and data center level.

Since the pioneering work in [46], power-aware task scheduling on processors with variable voltages and speeds has been extensively studied [6,11,17,22]. Performance constrained energy reduction in a computing system with multiple tasks was first studied in [48], and the research has been extended by a number of researchers in substantial further investigation [5,9,21,29–31,49]. Significant research has been focused on real-time applications, namely, adjusting the supply voltage and clock frequency to minimize processor energy consumption while still meeting the deadlines for task execution [4,15,16,20,23,32–34,36,37,47,54–57]. Energy and time constrained power allocation and task scheduling on multiprocessor computers with dynamically variable voltage and frequency and speed and power have also been addressed as combinatorial optimization problems [8,24–27,35].

Efficient power management and performance optimization in large-scale data centers and server clusters has gained much attention in the research community in recent years. In [18], the authors developed a framework for hierarchical autonomic power and performance management in high-performance distributed data centers. In [44], the authors proposed a highly scalable hierarchical power control architecture for large-scale data centers. In [45], the authors presented a novel cluster-level control architecture that coordinates individual power and performance control loops for virtualized server clusters. In [51–53], the authors formulated an optimization problem to get an optimal resource scheduling strategy for a given parallel workload in a server cluster, such that the proposed optimization model provides controllable and predictable quantitative control of power consumption with theoretically guaranteed service performance, which is essentially performance constrained power minimization.

Our investigation in this paper belongs to power constrained performance optimization at the data center level by formulating and solving a multivariable optimization problem. Such an approach has rarely been seen in the existing literature [28].

3. Power model

Power dissipation and circuit delay in digital CMOS circuits can be accurately modeled by simple equations, even for complex microprocessor circuits. CMOS circuits have dynamic, static, and short-circuit power dissipation; however, the dominant component in a well designed circuit is dynamic power consumption p (i.e., the switching component of power), which is approximately $p = aCV^2f$, where a is an activity factor, C is the loading capacitance, V is the supply voltage, and f is the clock frequency [10]. Since $s \propto f$, where s is the processor speed, and $f \propto V^{\phi}$ with $0 < \phi \leq 1$ [50], which implies that $V \propto f^{1/\phi}$, we know that power consumption is $p \propto f^{\alpha}$ and $p \propto s^{\alpha}$, where $\alpha = 1 + 2/\phi \geq 3$.

4. Performance model

Throughout the paper, we use $\overline{x}$ to denote the expectation of a random variable x and σ_x^2 to denote the variance of x and $c_x = \sigma_x/\overline{x}$ to denote the coefficient of variation of x.

Assume that we have n heterogeneous servers $1, 2, \ldots, n$ in a data center, each having its own arrival stream of tasks and power supply. There is no load distribution and balancing mechanism. A task submitted to a server must be processed on that server and task mitigation/migration/rejection is not allowed. System performance optimization is achieved by optimal power allocation.

Each server is modeled as an M/G/1 queueing system. Assume that there is a Poisson stream of arrival tasks to server i with arrival rate λ_i (measured in the number of tasks per second). Let $\lambda = \lambda_1 + \lambda_2 + \cdots + \lambda_n$ be the total arrival rate.

Let r_i represent the random execution requirement (measured in the number of giga instructions) of a task submitted to server i, where $1 \le i \le n$. Notice that r_i can have an arbitrary probability distribution. We use p_i to represent the power (measured in Watt) supplied to server i. For ease of discussion, we will assume that p_i is simply s_i^α, where $s_i = p_i^{1/\alpha}$ is the execution speed of server i (measured in the number of giga instructions executed per second). The random execution time of a task on server i is $t_i = r_i/s_i = r_i/p_i^{1/\alpha}$ (measured in second). Since t_i and r_i are linearly related, they have the same coefficient of variation $c_{t_i} = c_{r_i} = c_i$.

Let $\rho_i = \lambda_i \bar{t}_i = \lambda_i \bar{r}_i / p_i^{1/\alpha} = w_i / p_i^{1/\alpha}$ denote the utilization of server i, where $w_i = \lambda_i \bar{r}_i$ is the expected amount of work received by server i in a unit of time. Since $\rho_i < 1$, we must have $p_i > w_i^\alpha$.

By using the well known Pollaczek–Khinchin mean-value formula [19], we get the average task response time of server i, i.e.,

$$T_i = \bar{t}_i \left(1 + \frac{(1 + c_{t_i}^2)\rho_i}{2(1-\rho_i)} \right) = \frac{\bar{r}_i}{p_i^{1/\alpha}} \left(1 + \frac{(1+c_i^2)w_i}{2(p_i^{1/\alpha} - w_i)} \right).$$

The average task response time in the data center with n servers is

$$\begin{aligned} T(p_1, p_2, \ldots, p_n) &= \sum_{i=1}^{n} \left(\frac{\lambda_i}{\lambda} \right) T_i \\ &= \frac{1}{\lambda} \sum_{i=1}^{n} \frac{\lambda_i \bar{r}_i}{p_i^{1/\alpha}} \left(1 + \frac{(1+c_i^2)w_i}{2(p_i^{1/\alpha} - w_i)} \right) \\ &= \frac{1}{\lambda} \sum_{i=1}^{n} \frac{w_i}{p_i^{1/\alpha}} \left(1 + \frac{(1+c_i^2)w_i}{2(p_i^{1/\alpha} - w_i)} \right) \\ &= \frac{1}{\lambda} \sum_{i=1}^{n} w_i \left(\frac{1}{p_i^{1/\alpha}} + \frac{(1+c_i^2)w_i}{2(p_i^{2/\alpha} - w_i p_i^{1/\alpha})} \right), \end{aligned} \quad (1)$$

where we view T as a function of power supplies $p_1, p_2, \ldots, p_n$.

5. Problem formulation

Assume that an idle computer i consumes certain base power p_i^*, which includes static power dissipation, short circuit power dissipation, and other leakage and wasted power [1]. Given power supply p_i, the expected energy consumption of server i over a time period of τ is

$$\begin{aligned} e_i &= \tau(\rho_i p_i + p_i^*) \\ &= \tau(\lambda_i \bar{t}_i p_i + p_i^*) \\ &= \tau(\lambda_i(\bar{r}_i/p_i^{1/\alpha})p_i + p_i^*) \\ &= \tau(\lambda_i \bar{r}_i p_i^{1-1/\alpha} + p_i^*) \\ &= \tau(w_i p_i^{1-1/\alpha} + p_i^*), \end{aligned}$$

where $w_i p_i^{1-1/\alpha} + p_i^*$ is the effective power consumed by server i. The total expected energy consumption of the n servers is

$$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n} \tau(w_i p_i^{1-1/\alpha} + p_i^*) = \tau \sum_{i=1}^{n} (w_i p_i^{1-1/\alpha} + p_i^*),$$

where

$$\sum_{i=1}^{n} (w_i p_i^{1-1/\alpha} + p_i^*)$$

is the total effective power consumed by the n servers. The average task response time $T(p_1, p_2, \ldots, p_n)$ is minimized subject to the constraint that

$$\sum_{i=1}^{n} \tau(w_i p_i^{1-1/\alpha} + p_i^*) = E,$$

where E is a given energy constraint. The above condition is equivalent to

$$\sum_{i=1}^{n} (w_i p_i^{1-1/\alpha} + p_i^*) = \frac{E}{\tau},$$

where E/τ is a constraint on the total effective power. In other words, we have

$$F(p_1, p_2, \ldots, p_n) = \sum_{i=1}^{n} w_i p_i^{1-1/\alpha} = P, \quad (2)$$

where

$$P = \frac{E}{\tau} - \sum_{i=1}^{n} p_i^*,$$

is the total available effective power to be allocated. Notice that we must have $\rho_i < 1$, i.e., $p_i^{1/\alpha} > w_i$, or $p_i > w_i^\alpha$, for all $1 \le i \le n$, and

$$P = \sum_{i=1}^{n} w_i p_i^{1-1/\alpha} > \sum_{i=1}^{n} w_i^\alpha,$$

for all the n servers to run fast enough to handle the n task arrival streams.

Our optimization problem is defined as follows: given task arrival rates $\lambda_1, \lambda_2, \ldots, \lambda_n$, expected task execution requirements $\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_n$, variances of task execution requirements $\sigma_{r_1}^2, \sigma_{r_2}^2, \ldots, \sigma_{r_n}^2$, and total available effective power P, find optimal power supplies $p_1, p_2, \ldots, p_n$ which minimize the average task response time $T(p_1, p_2, \ldots, p_n)$ in (1) subject to the power constraint in (2). Notice that the objective of the optimization problem is to reduce the average response time of all the servers in a data center. These servers are entirely heterogeneous in terms of task arrival rate, task execution requirement, coefficient of variation of task execution requirements, base power consumption, power supply, task execution speed, server utilization, and task response time.

6. An algorithm for numerical solutions

We can minimize $T(p_1, p_2, \ldots, p_n)$ subject to the constraint $F(p_1, p_2, \ldots, p_n) = P$ by using the Lagrange multiplier system

$$\nabla T(p_1, p_2, \ldots, p_n) = y \nabla F(p_1, p_2, \ldots, p_n),$$

where y is a Lagrange multiplier. Notice that

$$\begin{aligned}
&\frac{\partial T(p_1, p_2, \ldots, p_n)}{\partial p_i}\\
&= \frac{w_i}{\lambda}\left(-\frac{1}{\alpha p_i^{1+1/\alpha}} - \frac{(1+c_i^2)w_i}{2}\cdot\frac{(2/\alpha)p_i^{2/\alpha-1} - w_i(1/\alpha)p^{1/\alpha-1}}{(p_i^{2/\alpha} - w_i p_i^{1/\alpha})^2}\right)\\
&= -\frac{w_i}{\lambda\alpha}\left(\frac{1}{p_i^{1+1/\alpha}} + \frac{(1+c_i^2)w_i}{2}\cdot\left(\frac{2}{p_i^{1-2/\alpha}} - \frac{w_i}{p_i^{1-1/\alpha}}\right)\cdot\frac{1}{(p_i^{2/\alpha} - w_i p_i^{1/\alpha})^2}\right)\\
&= -\frac{w_i}{\lambda\alpha}\left(\frac{1}{p_i^{1+1/\alpha}} + \frac{(1+c_i^2)w_i}{2}\cdot\frac{2p_i^{1/\alpha} - w_i}{p_i^{1-1/\alpha}}\cdot\frac{1}{(p_i^{2/\alpha} - w_i p_i^{1/\alpha})^2}\right)\\
&= -\frac{w_i}{\lambda\alpha p_i^{1+1/\alpha}}\left(1 + \frac{(1+c_i^2)w_i}{2}\cdot\frac{2p_i^{1/\alpha} - w_i}{(p_i^{1/\alpha} - w_i)^2}\right).
\end{aligned}$$

Also, we have

$$\frac{\partial F(p_1, p_2, \ldots, p_n)}{\partial p_i} = \left(1 - \frac{1}{\alpha}\right)\frac{w_i}{p_i^{1/\alpha}}.$$

Since

$$\frac{\partial T(p_1, p_2, \ldots, p_n)}{\partial p_i} = y\frac{\partial F(p_1, p_2, \ldots, p_n)}{\partial p_i},$$

we obtain

$$-\frac{w_i}{\lambda\alpha p_i^{1+1/\alpha}}\left(1 + \frac{(1+c_i^2)w_i}{2}\cdot\frac{2p_i^{1/\alpha} - w_i}{(p_i^{1/\alpha} - w_i)^2}\right) = y\left(1 - \frac{1}{\alpha}\right)\frac{w_i}{p_i^{1/\alpha}},$$

which can be simplified as

$$\frac{1}{\lambda p_i}\left(1 + \frac{(1+c_i^2)w_i}{2}\cdot\frac{2p_i^{1/\alpha} - w_i}{(p_i^{1/\alpha} - w_i)^2}\right) = (-y)(\alpha - 1). \qquad (3)$$

Thus, we have a nonlinear system of $n+1$ equations from (2) and (3).

A closed-form solution to Eq. (3) can be obtained for the special case when $c_i = 1$ (e.g., when server i is an M/M/1 queueing system) and $\alpha = 3$. In this case, Eq. (3) becomes

$$p_i^{1/3}(p_i^{1/3} - w_i)^2 = -\frac{1}{2\lambda y}.$$

Let $x_i = p_i^{1/3}$. Then, we get

$$x_i^3 - 2w_i x_i^2 + w_i^2 x_i + \frac{1}{2\lambda y} = 0.$$

The above cubic equation can be solved by using a standard method (see, e.g., p. 82 of [58]). If we let

$$z_i = x_i - \frac{2w_i}{3},$$

we have

$$z_i^3 + 3C_i z_i + D_i = 0,$$

where

$$C_i = -\frac{w_i^2}{9},$$

and

$$D_i = \frac{2w_i^3}{27} + \frac{1}{2\lambda y}.$$

If $4C_i^3 + D_i^2 > 0$, we get

$$z_i = \left(\frac{-D_i + \sqrt{4C_i^3 + D_i^2}}{2}\right)^{1/3} + \left(\frac{-D_i - \sqrt{4C_i^3 + D_i^2}}{2}\right)^{1/3}.$$

If $4C_i^3 + D_i^2 \le 0$, we get

$$z_i = 2(-C_i^3)^{1/6}\cos\frac{\theta_i}{3} = \frac{2}{3}w_i\cos\frac{\theta_i}{3},$$

where

$$\begin{aligned}
\theta_i &= \cos^{-1}\left(-\frac{D_i}{2\sqrt{-C_i^3}}\right) = \cos^{-1}\left(-\frac{27D_i}{2w_i^3}\right)\\
&= \cos^{-1}\left(-\left(1 + \frac{27}{4w_i^3\lambda y}\right)\right).
\end{aligned}$$

Based on z_i, we obtain

$$p_i = x_i^3 = \left(z_i + \frac{2}{3}w_i\right)^3,$$

for all $1 \le i \le n$. However, there is no closed-form solution for y.

It is unlikely that the above nonlinear system of equations accommodates a closed-form solution. We use the following strategy to find a numerical solution $(y, p_1, p_2, \ldots, p_n)$.

6.1. An algorithm for solving Eq. (3)

Our algorithm for finding a numerical solution $(y, p_1, p_2, \ldots, p_n)$ repeatedly uses a subroutine to solve Eq. (3). Given λ, c_i, w_i, and y, our algorithm to find $p_i \in (w_i^\alpha, \infty)$ which satisfies (3) is described as follows. Let $f(p_i)$ be the left-hand side of (3), i.e.,

$$f(p_i) = \frac{1}{\lambda p_i}\left(1 + \frac{(1+c_i^2)w_i}{2}\cdot\frac{2p_i^{1/\alpha} - w_i}{(p_i^{1/\alpha} - w_i)^2}\right).$$

It is easy to verify that $f(p_i)$ is a strictly decreasing function of p_i in the domain (w_i^α, ∞) with range $(0, \infty)$. Thus, there is a unique solution to p_i for arbitrary $y < 0$. Our idea is to find an interval $[\mathit{left}, \mathit{right})$ such that the solution to p_i can be found in $[\mathit{left}, \mathit{right})$ by using the bisection method with arbitrary precision.

The algorithm has three major steps.

Step 1. Our identification of the interval $[\mathit{left}, \mathit{right})$ starts with $p_i^{(0)} = 2w_i^\alpha$.

Step 2A. If $f(p_i^{(0)}) < (-y)(\alpha - 1)$, then $[\mathit{left}, \mathit{right}) \subset (w_i^\alpha, p_i^{(0)})$. We consider a sequence of p_i: $p_i^{(0)}, p_i^{(1)}, p_i^{(2)}, \ldots, p_i^{(m-1)}, p_i^{(m)}$, where the distance from $p_i^{(j)}$ to w_i^α is half of the distance from $p_i^{(j-1)}$ to w_i^α, i.e., $p_i^{(j)} - w_i^\alpha = (p_i^{(j-1)} - w_i^\alpha)/2$, or, $p_i^{(j)} = (p_i^{(j-1)} + w_i^\alpha)/2$, for all $1 \le j \le m$. The value of $m \ge 1$ is determined such that $f(p_i^{(m-1)}) < (-y)(\alpha - 1)$ but $f(p_i^{(m)}) \ge (-y)(\alpha - 1)$. The interval $[\mathit{left}, \mathit{right})$ is then $[p_i^{(m)}, p_i^{(m-1)}) = [p_i^{(m)}, 2p_i^{(m)} - w_i^\alpha)$.

Step 2B. If $f(p_i^{(0)}) \ge (-y)(\alpha - 1)$, then $[\mathit{left}, \mathit{right}) \subset [p_i^{(0)}, \infty)$. We consider a sequence of p_i: $p_i^{(0)}, p_i^{(1)}, p_i^{(2)}, \ldots, p_i^{(m-1)}, p_i^{(m)}$, where $p_i^{(j)} = 2p_i^{(j-1)}$, for all $1 \le j \le m$. The value of $m \ge 1$ is determined such that $f(p_i^{(m-1)}) \ge (-y)(\alpha - 1)$ but $f(p_i^{(m)}) < (-y)(\alpha - 1)$. The interval $[\mathit{left}, \mathit{right})$ is then $[p_i^{(m-1)}, p_i^{(m)}) = [p_i^{(m)}/2, p_i^{(m)})$.

Step 3. The length of the interval $[\mathit{left}, \mathit{right})$ is repeatedly reduced by half until it is no larger than some prespecified accuracy $\epsilon > 0$. The middle point $(\mathit{left} + \mathit{right})/2$ is returned as a numerical solution to p_i with accuracy ϵ.

The above algorithm is formally described in Fig. 1.

```
Algorithm 1: Solving Equation (3)
Input: $\lambda$, $c_i$, $w_i$, and $y$.
Output: $p_i \in (w_i^\alpha, \infty)$ which satisfies (3).

//Step 1
$p_i \leftarrow 2w_i^\alpha$
//Step 2A
if $f(p_i) < (-y)(\alpha - 1)$
    repeat
        $p_i \leftarrow (p_i + w_i^\alpha)/2$
    until $f(p_i) \geq (-y)(\alpha - 1)$
    left $\leftarrow p_i$
    right $\leftarrow 2p_i - w_i^\alpha$
//Step 2B
else
    repeat
        $p_i \leftarrow 2p_i$
    until $f(p_i) < (-y)(\alpha - 1)$
    left $\leftarrow p_i/2$
    right $\leftarrow p_i$
endif
//Step 3
repeat
    $p_i' \leftarrow p_i$
    $p_i \leftarrow (left + right)/2$
    if $f(p_i) \geq (-y)(\alpha - 1)$
        left $\leftarrow (left + right)/2$
    else
        right $\leftarrow (left + right)/2$
    endif
until $|p_i - p_i'| \leq \epsilon$
return $p_i$
```

Fig. 1. Algorithm for solving Eq. (3).

6.2. *An algorithm for finding $(y, p_1, p_2, \ldots, p_n)$*

Given λ, $w_1, w_2, \ldots, w_n$, c_1, c_2, …, c_n, (these data are from the input to our optimization problem, namely, λ_1, λ_2, …, λ_n, $\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_n$, $\sigma^2_{r_1}, \sigma^2_{r_2}, \ldots, \sigma^2_{r_n}$) and P, our algorithm to find a numerical solution $(y, p_1, p_2, \ldots, p_n)$ which satisfies (2) and (3) is described as follows. The algorithm has three major steps.

Step 1. We simplify (3) as

$$\frac{1}{\lambda p_i} = (-y^{(0)})(\alpha - 1),$$

that is,

$$p_i = \frac{1}{\lambda(-y^{(0)})(\alpha - 1)}.$$

Putting the p_i's into (2), we get

$$\sum_{i=1}^{n} \frac{w_i}{(\lambda(-y^{(0)})(\alpha - 1))^{1-1/\alpha}} = P,$$

which gives rise to

$$y^{(0)} = -\frac{1}{\lambda(\alpha - 1)} \left(\frac{1}{P} \sum_{i=1}^{n} w_i \right)^{\alpha/(\alpha-1)}.$$

By using $y^{(0)}$ and solving Eq. (3), we obtain $(p_1^{(0)}, p_2^{(0)}, \ldots, p_n^{(0)})$. Unfortunately, Eq. (3) yields $(p_1^{(0)}, p_2^{(0)}, \ldots, p_n^{(0)})$ which are too large, since

$$\sum_{i=1}^{n} w_i (p_i^{(0)})^{1-1/\alpha} > P.$$

Step 2. We consider a sequence of y: $y^{(0)}, y^{(1)}, y^{(2)}, \ldots, y^{(m-1)}, y^{(m)}$, where $y^{(j)} = 2y^{(j-1)}$ for all $j = 1, 2, \ldots, m$, and each $y^{(j)}$ results in $(p_1^{(j)}, p_2^{(j)}, \ldots, p_n^{(j)})$ by solving Eq. (3). The sequence of decreased value of y give decreased $(p_1^{(j)}, p_2^{(j)}, \ldots, p_n^{(j)})$, because the left-hand side of (3) is a strictly decreasing function of p_i. The value $m \geq 1$ is determined such that

$$\sum_{i=1}^{n} w_i (p_i^{(m-1)})^{1-1/\alpha} > P,$$

but

$$\sum_{i=1}^{n} w_i (p_i^{(m)})^{1-1/\alpha} \leq P.$$

Step 3. We search y in the interval $[\mathit{left}, \mathit{right}) = [y^{(m)}, y^{(m-1)}) = [y^{(m)}, y^{(m)}/2)$ by using the bisection method such that the resulted $(y, p_1, p_2, \ldots, p_n)$ are all within some prespecified accuracy $\epsilon > 0$.

The above algorithm is formally described in Fig. 2.

7. Numerical data

The purpose of this section is to demonstrate numerical data. The significance of these data is to show the impact of the task arrival rates, the expected task execution requirements, and the coefficients of variation of task execution requirements on the average task response time. All our parameters are chosen in such a way that in each figure, we can show the impact of one of the major factors, i.e., the task arrival rates, the expected task execution requirements, and the coefficients of variation of task execution requirements.

We consider a data center containing $n = 10$ heterogeneous servers with $\alpha = 3$.

Let the unit of time be normalized such that the minimum arrival rate is one, and $\lambda_i = 1 + l(i-1)$ for all $1 \leq i \leq n$. Let the measure of task execution requirement be normalized such that the minimum requirement is one, and $r_i = 1.0 + 0.1(i-1)$ for all $1 \leq i \leq n$. We set $\sigma_{r_i} = 0.5 + 0.2(i-1)$ for all $1 \leq i \leq n$. In Fig. 3, we show the average task response time T vs. total effective power P and l, where $l = 0.100, 0.125, 0.150, 0.175, 0.200$. All the data are calculated by using Algorithms 1 and 2 with $\epsilon = 10^{-10}$.

As another example, we set $\lambda = 1 + 0.1(i-1)$, $r_i = 1 + u(i-1)$, and $\sigma_{r_i} = 0.5 + 0.2(i-1)$, for all $1 \leq i \leq n$. In Fig. 4, we show the average task response time T vs. total effective power P and u, where $u = 0.100, 0.125, 0.150, 0.175, 0.200$.

As a third example, we set $\lambda = 1 + 0.1(i-1)$, $r_i = 1 + 0.1(i-1)$, and $\sigma_{r_i} = 0.5 + v(i-1)$, for all $1 \leq i \leq n$. In Fig. 5, we show the average task response time T vs. total effective power P and v, where $v = 0.20, 0.50, 0.80, 0.11, 0.14$.

The values of l, u, v are selected in such a way that the impact of the task arrival rates, the expected task execution requirements, and the coefficients of variation of task execution requirements can be clearly demonstrated. We observe that the task arrival rates, the expected task execution requirements, and the coefficients of variation of task execution requirements all have significant impact on the average task response time in a data center.

8. Heuristic solutions

A number of heuristic solutions can be developed for the optimal power allocation problem. These methods provide simple and even closed-form solutions which yield the average task response time very close to the optimal solution. The benefit of a heuristic method is that it can provide a quick yet accurate solution.

8.1. The workload proportional method

In the *workload proportional* (WP) method, also called the energy proportional method [53], the power allocated to a server is proportional to the workload on the server. Hence, for n servers with workloads $w_1, w_2, \ldots, w_n$, we have

$$p_i = x\left(\frac{w_i}{w_1 + w_2 + \cdots + w_n}\right)P,$$

for some x, where $1 \leq i \leq n$. Notice that

$$\sum_{i=1}^{n} w_i p_i^{1-1/\alpha} = P,$$

that is,

$$x^{1-1/\alpha} P^{1-1/\alpha} \sum_{i=1}^{n} w_i \left(\frac{w_i}{w_1 + w_2 + \cdots + w_n}\right)^{1-1/\alpha} = P,$$

or, equivalently,

$$x^{1-1/\alpha}\left(\frac{w_1^{2-1/\alpha} + w_2^{2-1/\alpha} + \cdots + w_n^{2-1/\alpha}}{(w_1 + w_2 + \cdots + w_n)^{1-1/\alpha}}\right) = P^{1/\alpha},$$

and

$$x^{\alpha-1}\left(\frac{(w_1^{2-1/\alpha} + w_2^{2-1/\alpha} + \cdots + w_n^{2-1/\alpha})^{\alpha}}{(w_1 + w_2 + \cdots + w_n)^{\alpha-1}}\right) = P.$$

The last equation gives rise to

$$x = \left(\frac{w_1 + w_2 + \cdots + w_n}{(w_1^{2-1/\alpha} + w_2^{2-1/\alpha} + \cdots + w_n^{2-1/\alpha})^{\alpha/(\alpha-1)}}\right) P^{1/(\alpha-1)},$$

and

$$p_i = \left(\frac{w_i}{(w_1^{2-1/\alpha} + w_2^{2-1/\alpha} + \cdots + w_n^{2-1/\alpha})^{\alpha/(\alpha-1)}}\right) P^{\alpha/(\alpha-1)},$$

for all $1 \leq i \leq n$.

8.2. The equal speed method

Another workload proportional method is that the effective power consumed by a server is proportional to the workload on the server, namely,

$$w_i p_i^{1-1/\alpha} = \left(\frac{w_i}{w_1 + w_2 + \cdots + w_n}\right)P,$$

which implies that

$$p_i = \left(\frac{P}{w_1 + w_2 + \cdots + w_n}\right)^{\alpha/(\alpha-1)},$$

for all $1 \leq i \leq n$. The above equation indicates that all servers have the same power and speed. Hence, we call this method as the *equal speed* (ES) method.

```
Algorithm 2: Finding $(y, p_1, p_2, \ldots, p_n)$
Input: $\lambda, w_1, w_2, \ldots, w_n, c_1, c_2, \ldots, c_n$, and $P$
Output: $y, p_1, p_2, \ldots, p_n$

//Step 1
$y \leftarrow -\frac{1}{\lambda(\alpha-1)}\left(\frac{1}{P}\sum_{i=1}^{n} w_i\right)^{\alpha/(\alpha-1)}$
//Step 2
repeat
    $y \leftarrow 2y$
    for $(i = 1; i \le n; i{+}{+})$
        find $p_i$ using Algorithm 1
until $\sum_{i=1}^{n} w_i p_i^{1-1/\alpha} \le P$
//Step 3
$left \leftarrow y$
$right \leftarrow y/2$
repeat
    $(y', p'_1, p'_2, \ldots, p'_n) \leftarrow (y, p_1, p_2, \ldots, p_n)$
    $y \leftarrow (left + right)/2$
    for $(i = 1; i \le n; i{+}{+})$
        find $p_i$ using Algorithm 1
    if $\sum_{i=1}^{n} w_i p_i^{1-1/\alpha} \le P$
        $left \leftarrow (left + right)/2$
    else
        $right \leftarrow (left + right)/2$
    endif
until $\max(|y - y'|, |p_1 - p'_1|, |p_2 - p'_2|, \ldots, |p_n - p'_n|) \le \epsilon$
return $(y, p_1, p_2, \ldots, p_n)$
```

Fig. 2. Algorithm for finding $(y, p_1, p_2, \ldots, p_n)$.

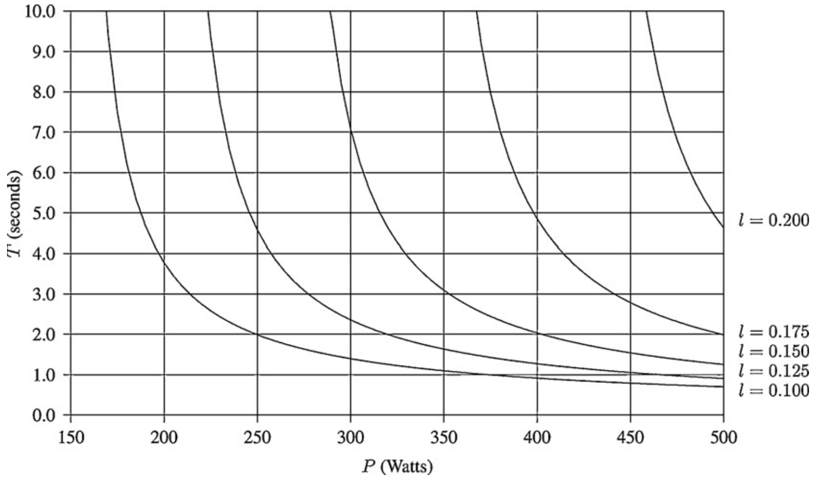

Fig. 3. The average task response time T vs. total effective power P and l.

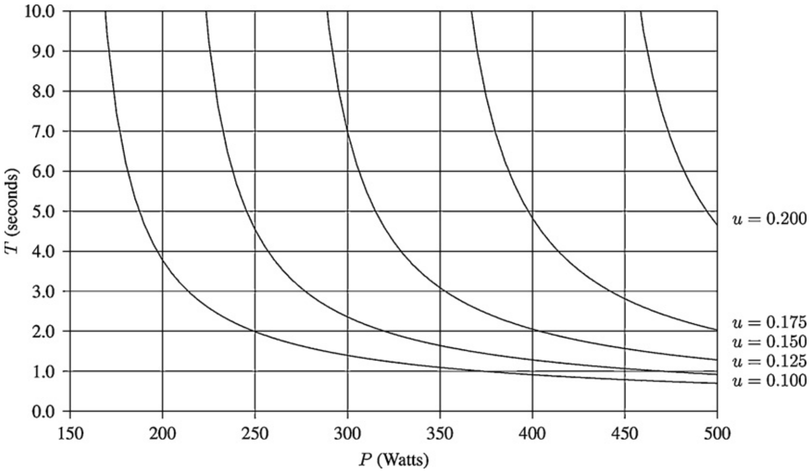

Fig. 4. The average task response time T vs. total effective power P and u.

8.3. *The equal utilization method*

In the *equal utilization* (EU) method, the available power P is allocated to the servers in such a way that all the servers have the same utilization, i.e., $\rho_1 = \rho_2 = \cdots = \rho_n = \rho$. Since

$$\rho_i = \frac{w_i}{p_i^{1/\alpha}} = \rho,$$

we have

$$p_i = \left(\frac{w_i}{\rho}\right)^{\alpha},$$

for all $1 \le i \le n$. Based on the condition that

$$\sum_{i=1}^{n} w_i p_i^{1-1/\alpha} = P,$$

we get

$$P = \sum_{i=1}^{n} w_i \left(\frac{w_i}{\rho}\right)^{\alpha-1} = \frac{1}{\rho^{\alpha-1}} \sum_{i=1}^{n} w_i^{\alpha},$$

which implies that

$$\rho^{\alpha-1} = \frac{1}{P} \sum_{i=1}^{n} w_i^{\alpha},$$

and

$$p_i = \left(\frac{w_i^{\alpha}}{(w_1^{\alpha} + w_2^{\alpha} + \cdots + w_n^{\alpha})^{\alpha/(\alpha-1)}}\right) P^{\alpha/(\alpha-1)},$$

for all $1 \le i \le n$.

8.4. *The equal time method*

In the *equal time* (ET) method, the available power P is allocated to the servers in such a way that all the servers have the same average task response time, i.e., $T_1 = T_2 = \cdots = T_n = T$, which is also the average task response time in a data center with n servers. Since

$$T_i = \frac{\bar{r}_i}{p_i^{1/\alpha}} \left(1 + \frac{(1 + c_i^2) w_i}{2(p_i^{1/\alpha} - w_i)}\right) = T,$$

we obtain

$$2T p_i^{2/\alpha} - 2(\bar{r}_i + w_i T) p_i^{1/\alpha} - \bar{r}_i w_i (c_i^2 - 1) = 0,$$

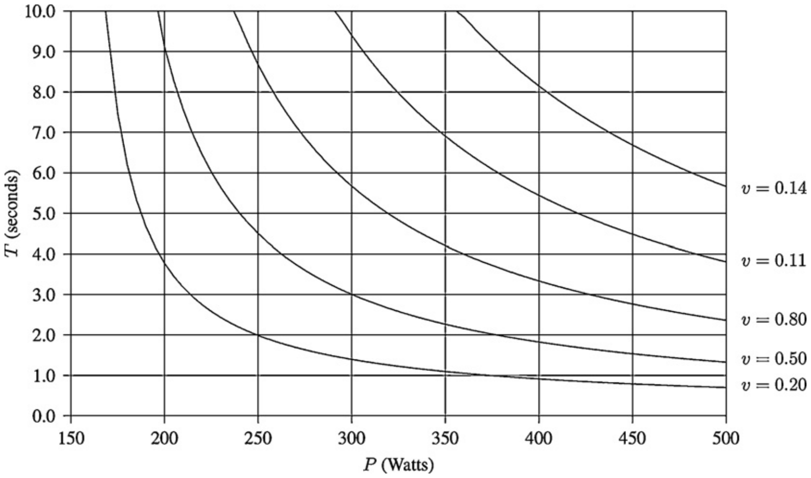

Fig. 5. The average task response time T vs. total effective power P and v.

Table 1
Comparison of heuristic solutions.

P	ES	WP	EU	ET	OPT
160	–	–	23.6464820	23.6014442	20.0403962
170	–	–	11.0667775	10.9992702	9.3999089
180	–	–	7.3041328	7.2317756	6.2144186
190	–	–	5.4923905	5.4189046	4.6790143
200	–	–	4.4255392	4.3522292	3.7739411
210	–	–	3.7218538	3.6492758	3.1763537
220	–	–	3.2224276	3.1508370	2.7518160
230	–	5.9323698	2.8492928	2.7788051	2.4343428
240	–	3.7009223	2.5597051	2.4903646	2.1877444
250	–	2.8692808	2.3282689	2.2600817	1.9905099
260	–	2.4058969	2.1389450	2.0718960	1.8290474
270	–	2.1003346	1.9811038	1.9151662	1.6943447
280	–	1.8793819	1.8474213	1.7825620	1.5801894
290	10.6952124	1.7100732	1.7326864	1.6688692	1.4821590
300	4.5150157	1.5750692	1.6330896	1.5702771	1.3970189
310	3.1538259	1.4642476	1.5457811	1.4839361	1.3223482
320	2.5251252	1.3712394	1.4685866	1.4076725	1.2562991
330	2.1507320	1.2918056	1.3998178	1.3397991	1.1974358
340	1.8967079	1.2229978	1.3381438	1.2789864	1.1446263
350	1.7102047	1.1626919	1.2825011	1.2241724	1.0969655
360	1.5658943	1.1093131	1.2320298	1.1744987	1.0537211
370	1.4499911	1.0616653	1.1860266	1.1292637	1.0142942
380	1.3542846	1.0188216	1.1439110	1.0878882	0.9781900
390	1.2735451	0.9800508	1.1051995	1.0498902	0.9449963
400	1.2042637	0.9447673	1.0694859	1.0148649	0.9143669
410	1.1439863	0.9124961	1.0364268	0.9824702	0.8860085
420	1.0909381	0.8828470	1.0057296	0.9524145	0.8596715
430	1.0437989	0.8554966	0.9771432	0.9244484	0.8351418
440	1.0015634	0.8301738	0.9504517	0.8983565	0.8122346
450	0.9634508	0.8066497	0.9254673	0.8739524	0.7907897
460	0.9288437	0.7847296	0.9020268	0.8510738	0.7706677
470	0.8972467	0.7642467	0.8799873	0.8295787	0.7517463
480	0.8682570	0.7450571	0.8592231	0.8093422	0.7339180
490	0.8415434	0.7270359	0.8396232	0.7902541	0.7170880
500	0.8168303	0.7100742	0.8210893	0.7722168	0.7011721

for all $1 \le i \le n$. Solving the above quadratic equation of $p_i^{1/\alpha}$, we get

$$p_i^{1/\alpha} = \frac{\bar{r}_i + w_i T + \sqrt{(\bar{r}_i + w_i T)^2 + 2\bar{r}_i w_i (c_i^2 - 1)T}}{2T},$$

and

$$p_i = \left(\frac{\bar{r}_i + w_i T + \sqrt{(\bar{r}_i + w_i T)^2 + 2\bar{r}_i w_i (c_i^2 - 1)T}}{2T} \right)^{\alpha},$$

for all $1 \le i \le n$. Based on the condition that

$$\sum_{i=1}^{n} w_i p_i^{1-1/\alpha} = P,$$

we have

$$\sum_{i=1}^{n} w_i \left(\frac{\bar{r}_i + w_i T + \sqrt{(\bar{r}_i + w_i T)^2 + 2\bar{r}_i w_i (c_i^2 - 1)T}}{2T} \right)^{\alpha - 1} = P.$$

Notice that the left-hand side of the above equation is a decreasing function of T. Hence, given P, we can find T easily by using the bisection method.

8.5. Performance comparison

Again, let us consider a data center having $n = 10$ heterogeneous servers with $\alpha = 3$. In Table 1, we compare the average task response time T produced by the four heuristic solutions and the optimal solution, where we set $\lambda = 1 + 0.1(i - 1)$, $r_i = 1 + 0.1(i - 1)$, and $\sigma_{r_i} = 0.5 + 0.2(i - 1)$, for all $1 \le i \le n$. An entry with no datum means that the heuristic solution does not provide a meaningful solution (i.e., $p_i > w_i^{\alpha}$, for all $1 \le i \le n$).

We have a number of observations.

- All our heuristic solutions are reasonably or very close to the optimal solution OPT, except when a heuristic solution is close to its saturation point (i.e., when the task response time starts to grow sharply).
- The ES method performs consistently worse than other methods, because the equal speed method does not consider the workload on the servers and does not yield a good solution.
- The ET method performs consistently better than the EU method, because the equal time method attempts to balance the performance of all the servers.
- The WP method performs worse than EU and ET when P is small and WP is close to its saturation point. However, as P increases, WP performs better than EU and ET.

9. Summary

We have introduced the problem of optimal power allocation among multiple heterogeneous servers in a data center for cloud computing. The purpose is to provide the best quality of service by using certain limited power resource. Our approach is to model a server as an M/G/1 queueing system and formulate the average task response time in a data center with multiple servers as a function of power allocations to the servers. We have developed an algorithm to find the optimal solution numerically. We have also developed several closed-form heuristic solutions which are able to provide near-optimal solutions. Our approach provides an analytical way of studying the power-performance tradeoff at the data center level.

Notice that our power consumption model in this paper assumes that clock frequency and supply voltage and execution speed and power supply of a server can change continuously and unboundedly. However, in the current processor technology, clock frequency and supply voltage and execution speed and power supply can only be set with a few discrete levels. It is definitely interesting and important to formulate and solve our optimal power allocation problem for multiple heterogeneous servers with discrete and bounded and different clock frequency and supply voltage and execution speed and power supply levels. Such investigation will be practically more useful. Another direction worth of further investigation is to extend our work in this paper to more general queueing models such as G/G/1, which can be applied to more general servers, data centers, and cloud computing environments.

Acknowledgment

A preliminary version of the paper entitled "Performance optimization with energy constraint in heterogeneous multiple computer systems" was presented on *Workshop on Parallel Computing and Optimization*, Anchorage, Alaska, 2011.

References

[1] http://en.wikipedia.org/wiki/CMOS.

[2] http://en.wikipedia.org/wiki/Dedicated_hosting_service.

[3] S. Albers, Energy-efficient algorithms, Communications of the ACM 53 (5) (2010) 86–96.

[4] H. Aydin, R. Melhem, D. Mossé, P. Mejía-Alvarez, Power-aware scheduling for periodic real-time tasks, IEEE Transactions on Computers 53 (5) (2004) 584–600.

[5] N. Bansal, T. Kimbrel, K. Pruhs, Dynamic speed scaling to manage energy and temperature, in: Proceedings of the 45th IEEE Symposium on Foundation of Computer Science, 2004, pp. 520–529.

[6] J.A. Barnett, Dynamic task-level voltage scheduling optimizations, IEEE Transactions on Computers 54 (5) (2005) 508–520.

[7] L. Benini, A. Bogliolo, G. De Micheli, A survey of design techniques for system-level dynamic power management, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 8 (3) (2000) 299–316.

[8] D.P. Bunde, Power-aware scheduling for makespan and flow, in: Proceedings of the 18th ACM Symposium on Parallelism in Algorithms and Architectures, 2006, pp. 190–196.

[9] H.-L. Chan, W.-T. Chan, T.-W. Lam, L.-K. Lee, K.-S. Mak, P.W.H. Wong, Energy efficient online deadline scheduling, in: Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms, 2007, pp. 795–804.

[10] A.P. Chandrakasan, S. Sheng, R.W. Brodersen, Low-power CMOS digital design, IEEE Journal on Solid-State Circuits 27 (4) (1992) 473–484.

[11] S. Cho, R.G. Melhem, On the interplay of parallelization, program performance, and energy consumption, IEEE Transactions on Parallel and Distributed Systems 21 (3) (2010) 342–353.

[12] W.-C. Feng, The importance of being low power in high performance computing, CTWatch Quarterly, vol. 1, no. 3, Los Alamos National Laboratory, August 2005.

[13] A. Gara, al. et, Overview of the Blue Gene/L system architecture, IBM Journal of Research and Development 49 (2/3) (2005) 195–212.

[14] S.L. Graham, M. Snir, C.A. Patterson (Eds.), Getting Up to Speed: The Future of Supercomputing, Committee on the Future of Supercomputing, National Research Council, National Academies Press, 2005.

[15] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, M.B. Srivastava, Power optimization of variable-voltage core-based systems, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 18 (12) (1999) 1702–1714.

[16] C. Im, S. Ha, H. Kim, Dynamic voltage scheduling with buffers in low-power multimedia applications, ACM Transactions on Embedded Computing Systems 3 (4) (2004) 686–705.

[17] S.U. Khan, I. Ahmad, A cooperative game theoretical technique for joint optimization of energy consumption and response time in computational grids, IEEE Transactions on Parallel and Distributed Systems 20 (3) (2009) 346–360.

[18] B. Khargharia, S. Hariri, F. Szidarovszky, M. Houri, H. El-Rewini, S. Khan, I. Ahmad, M.S. Yousif, Autonomic power and performance management for large-scale data centers, NFS Next Generation Software Program (2007).

[19] L. Kleinrock, Queueing Systems, Volume 1: Theory, John Wiley and Sons, New York, 1975.

[20] C.M. Krishna, Y.-H. Lee, Voltage-clock-scaling adaptive scheduling techniques for low power in hard real-time systems, IEEE Transactions on Computers 52 (12) (2003) 1586–1593.

[21] W.-C. Kwon, T. Kim, Optimal voltage allocation techniques for dynamically variable voltage processors, ACM Transactions on Embedded Computing Systems 4 (1) (2005) 211–230.

[22] Y.C. Lee, A.Y. Zomaya, Energy conscious scheduling for distributed computing systems under different operating conditions, IEEE Transactions on Parallel and Distributed Systems 22 (8) (2011) 1374–1381.

[23] Y.-H. Lee, C.M. Krishna, Voltage-clock scaling for low energy consumption in fixed-priority real-time systems, Real-Time Systems 24 (3) (2003) 303–317.

[24] K. Li, Performance analysis of power-aware task scheduling algorithms on multiprocessor computers with dynamic voltage and speed, IEEE Transactions on Parallel and Distributed Systems 19 (11) (2008) 1484–1497.

[25] K. Li, Energy efficient scheduling of parallel tasks on multiprocessor computers, Journal of Supercomputing, doi:10.1007/s11227-010-0416-0, published online 12 March 2010.

[26] K. Li, Power allocation and task scheduling on multiprocessor computers with energy and time constraints, in: Y.-C. Lee, A. Zomaya (Eds.), Energy Aware Distributed Computing Systems, Wiley Series on Parallel and Distributed Computing, 2011.

[27] K. Li, Algorithms and analysis of energy-efficient scheduling of parallel tasks, in: S. Ranka, I. Ahmad (Eds.), Handbook of Energy-Aware and Green Computing, Chapman and Hall/CRC Press, 2011.

[28] K. Li, Performance optimization with energy constraint in heterogeneous multiple computer systems, Workshop on Parallel Computing and Optimization (2011), Anchorage, Alaska.

[29] M. Li, B.J. Liu, F.F. Yao, Min-energy voltage allocation for tree-structured tasks, Journal of Combinatorial Optimization 11 (2006) 305–319.

[30] M. Li, A.C. Yao, F.F. Yao, Discrete and continuous min-energy schedules for variable voltage processors, Proceedings of the National Academy of Sciences USA 103 (11) (2006) 3983–3987.

[31] M. Li, F.F. Yao, An efficient algorithm for computing optimal discrete voltage schedules, SIAM Journal on Computing 35 (3) (2006) 658–671.

[32] J.R. Lorch, A.J. Smith, PACE: a new approach to dynamic voltage scaling, IEEE Transactions on Computers 53 (7) (2004) 856–869.

[33] R.N. Mahapatra, W. Zhao, An energy-efficient slack distribution technique for multimode distributed real-time embedded systems, IEEE Transactions on Parallel and Distributed Systems 16 (7) (2005) 650–662.

[34] G. Quan, X.S. Hu, Energy efficient DVS schedule for fixed-priority real-time systems, ACM Transactions on Embedded Computing Systems 6 (4) (2007), Article no. 29.

[35] C. Rusu, R. Melhem, D. Mossé, Maximizing the system value while satisfying time and energy constraints, in: Proceedings of the 23rd IEEE Real-Time Systems Symposium, 2002, pp. 256–265.

[36] D. Shin, J. Kim, Power-aware scheduling of conditional task graphs in real-time multiprocessor systems, in: Proceedings of the International Symposium on Low Power Electronics and Design, 2003, pp. 408–413.

[37] D. Shin, J. Kim, S. Lee, Intra-task voltage scheduling for low-energy hard real-time applications, IEEE Design & Test of Computers 18 (2) (2001) 20–30.

[38] S. Srinivasan, V. Getov, Navigating the cloud computing landscape – technologies, services, and adopters, IEEE Computer 44 (3) (2011) 22–23.

[39] M.B. Srivastava, A.P. Chandrakasan, R.W. Rroderson, Predictive system shutdown and other architectural techniques for energy efficient programmable computation, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 4 (1) (1996) 42–55.

[40] M.R. Stan, K. Skadron, Guest editors' introduction: power-aware computing, IEEE Computer 36 (12) (2003) 35–38.

[41] O.S. Unsal, I. Koren, System-level power-aware design techniques in real-time systems, Proceedings of the IEEE 91 (7) (2003) 1055–1069.

[42] US EPA, Report to congress on server and data center energy efficiency, 2007.

[43] V. Venkatachalam, M. Franz, Power reduction techniques for microprocessor systems, ACM Computing Surveys 37 (3) (2005) 195–237.

[44] X. Wang, M. Chen, C. Lefurgy, T.W. Keller, SHIP: scalable hierarchical power control for large-scale data centers, in: Proceedings of the 18th International Conference on Parallel Architectures and Compilation Techniques, 2009, pp. 91–100.

[45] X. Wang, Y. Wang, Coordinating power control and performance management for virtualized server clusters, IEEE Transactions on Parallel and Distributed Systems, to appear (2011).

[46] M. Weiser, B. Welch, A. Demers, S. Shenker, Scheduling for reduced CPU energy, in: Proceedings of the 1st USENIX Symposium on Operating Systems Design and Implementation, 1994, pp. 13–23.

[47] P. Yang, C. Wong, P. Marchal, F. Catthoor, D. Desmet, D. Verkest, R. Lauwereins, Energy-aware runtime scheduling for embedded-multiprocessor SOCs, IEEE Design & Test of Computers 18 (5) (2001) 46–58.

[48] F. Yao, A. Demers, S. Shenker, A scheduling model for reduced CPU energy, in: Proceedings of the 36th IEEE Symposium on Foundations of Computer Science, 1995, pp. 374–382.

[49] H.-S. Yun, J. Kim, On energy-optimal voltage scheduling for fixed-priority hard real-time systems, ACM Transactions on Embedded Computing Systems 2 (3) (2003) 393–430.

[50] B. Zhai, D. Blaauw, D. Sylvester, K. Flautner, Theoretical and practical limits of dynamic voltage scaling, in: Proceedings of the 41st Design Automation Conference, 2004, pp. 868–873.

[51] X. Zheng, Y. Cai, Optimal server provisioning and frequency adjustment in server clusters, in: 39th International Conference on Parallel Processing Workshops, 2010, pp. 504–511.

[52] X. Zheng, Y. Cai, Optimal server allocation and frequency modulation on multi-core based server clusters, International Journal of Green Computing 1 (2) (2010) 18–30.

[53] X. Zheng, Y. Cai, Achieving energy proportionality in server clusters, International Journal of Computer Networks 1 (2) (2010) 21–35.

[54] X. Zhong, C.-Z. Xu, Energy-aware modeling and scheduling for dynamic voltage scaling with statistical real-time guarantee, IEEE Transactions on Computers 56 (3) (2007) 358–372.

[55] D. Zhu, R. Melhem, B.R. Childers, Scheduling with dynamic voltage/speed adjustment using slack reclamation in multiprocessor real-time systems, IEEE Transactions on Parallel and Distributed Systems 14 (7) (2003) 686–700.

[56] D. Zhu, D. Mossé, R. Melhem, Power-aware scheduling for AND/OR graphs in real-time systems, IEEE Transactions on Parallel and Distributed Systems 15 (9) (2004) 849–864.

[57] J. Zhuo, C. Chakrabarti, Energy-efficient dynamic task scheduling algorithms for DVS systems, ACM Transactions on Embedded Computing Systems 7 (2) (2008), Article no. 17.

[58] D. Zwillinger (Ed.), Standard Mathematical Tables and Formulae, 30th ed., CRC Press, Boca Raton, FL, 1996.

Keqin Li is a SUNY Distinguished Professor in computer science. His research interests are mainly in design and analysis of algorithms, parallel and distributed computing, and computer networking. He has contributed extensively to processor allocation and resource management; design and analysis of sequential/parallel, deterministic/probabilistic, and approximation algorithms; parallel and distributed computing systems performance analysis, prediction, and evaluation; job scheduling, task dispatching, and load balancing in heterogeneous distributed systems; dynamic tree embedding and randomized load distribution in static networks; parallel computing using optical interconnections; dynamic location management in wireless communication networks; routing and wavelength assignment in WDM optical networks; energy-efficient power management and performance optimization. His current research interests include lifetime maximization in sensor networks, file sharing in peer-to-peer systems, and cloud computing. He has published over 235 journal articles, book chapters, and research papers in refereed international conference proceedings. He has received several Best Paper Awards for his highest quality work. He is currently on the editorial board of *IEEE Transactions on Parallel and Distributed Systems*, *Journal of Parallel and Distributed Computing*, *International Journal of Parallel, Emergent and Distributed Systems*, *International Journal of High Performance Computing and Networking*, and *Optimization Letters*.