This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2706425, IEEE Transactions on Sustainable Computing

Optimal Task Dispatching on Multiple Heterogeneous Multiserver Systems with Dynamic Speed and Power Management

Keqin Li, *Fellow, IEEE*

Abstract—Cloud load balancing is the process of distributing workloads across multiple computing resources in a cloud environment. Load distribution in cloud computing systems is more challenging than in other systems. The purpose of the paper is to address the issue of optimal task dispatching on multiple heterogeneous multiserver systems with dynamic speed and power management. The main contributions of the paper are to solve three problems, i.e., the optimal task dispatching problems with minimized average task response time, minimized average power consumption, and minimized average cost-performance ratio, for multiple heterogeneous multiserver systems with dynamic d-speed and power management. In our study, multiserver systems with dynamic speed and power management are modeled as queueing systems, so that fundamental performance and cost metrics such as the average task response time and the average power consumption can be obtained analytically. Our research problems are formulated as multi-variable optimization problems and solved numerically. To the best of our knowledge, this is the first work that addresses load distribution for performance optimization, power minimization, and cost-performance ratio optimization, collectively on multiple heterogeneous servers with dynamic speed and power management.

Index Terms—Cost-performance ratio, dynamic speed and power management, multiserver system, power consumption, queueing model, response time, task dispatching.

1 Introduction

1.1 Motivation

CLOUD load balancing is the process of distributing workloads across multiple computing resources in a cloud environment [1]. Load balancing allows enterprises to manage application demands by allocating workload among multiple computers or servers [3]. Load distribution has been a classic research problem in distributed computing, cluster computing, and grid computing [36], and continues to be a fundamental issue in cloud computing, to effectively increase the quality of service to cloud users and to enhance the utilization of resources in cloud systems.

Load distribution in cloud computing systems is more challenging than in other systems due to several reasons. First, energy consumption has become a key issue for the normal operation and maintenance of cloud computing platforms and datacenters, raising serious concerns from cloud providers (see [7], [25], [31] for recent research on green data centers, cloud computing systems, and distributed systems), and load balancing becomes more difficult when reducing energy consumption is also taken into consideration. Second, modern servers deployed in cloud computing have become more and more sophisticated due to the multicore processor architectures, the technique of workload dependent dynamic power management [28], and heterogeneous servers which are different in computing capacity and capability, power consumption model, and dynamic speed and power management scheme. Third, the objective of traditional load distribution is essentially to reduce the average task response time (i.e., to increase the quality of service); however, in cloud computing, there are diversified objectives such as to reduce energy consumption (i.e., to decrease the cost of service) and to optimize the cost-performance ratio.

- *K. Li is with the Department of Computer Science, State University of New York, New Paltz, New York 12561, USA.*
E-mail: lik@newpaltz.edu

1.2 Our Contributions

The purpose of the paper is to address the issue of optimal task dispatching on multiple heterogeneous multiserver systems with dynamic speed and power management. The main contributions of the paper are to solve three problems, i.e., the optimal task dispatching problems with minimized average task response time, minimized average power consumption, and minimized average cost-performance ratio, for multiple heterogeneous multiserver systems with dynamic d-speed and power management. In our study, multiserver systems with dynamic speed and power management are modeled as queueing systems, so that fundamental performance and cost metrics such as the average task response time and the average power consumption can be obtained analytically (Sections 2–5). Our research problems are formulated as multi-variable optimization problems and solved numerically (Sections 6–8). To the best of our knowledge, this is the first work that addresses load distribution for performance optimization, power minimization, and cost-performance ratio optimization, collectively on multiple heterogeneous multiservers with dynamic speed and power management.

The rest of the paper is organized as follows. In Section 2, we characterize a multiserver system using a queueing model. In Section 3, we describe our server speed and power consumption models. In Section 4, we characterize a dynamic speed and power management scheme using a birth-death process. In Section 5, we consider the class of d-speed schemes. In Sections 6–8, we define

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2706425, IEEE Transactions on Sustainable Computing

and solve the three optimization problems respectively, present numerical data, and conduct performance comparison. In Section 9, we review related research in cloud load balancing. In Section 10, we conclude the paper.

2 Multiserver Systems

To formulate and study the problem of optimal task dispatching and load distribution for multiple heterogeneous multiserver systems with dynamic speed and power management in a cloud computing environment, we need an analytical model for a multiserver system. A queueing model for a group of n heterogeneous multiserver systems S_1, S_2, ..., S_n of sizes m_1, m_2, ..., m_n and speeds s_1, s_2, ..., s_n will be employed in this paper. Assume that a multiserver system S_i has m_i identical servers with speed s_i. Such a multiserver system can be treated as an M/M/m queueing system which is elaborated as follows.

There is a Poisson stream of tasks with arrival rate λ (measured by the number of tasks per second), i.e., the inter-arrival times are independent and identically distributed (i.i.d.) exponential random variables with mean $1/\lambda$. A task dispatching and load distribution algorithm splits the stream into n substreams, such that the ith substream with arrival rate λ_i is sent to multiserver system S_i, where $1 \le i \le n$, and $\lambda = \lambda_1 + \lambda_2 + \cdots + \lambda_n$. A multiserver system S_i maintains a queue with infinite capacity for waiting tasks when all its m_i servers are busy. The first-come-first-served (FCFS) queueing discipline is adopted by all multiserver systems. The task execution requirements (measured by the number of billion instructions to be executed) are i.i.d. exponential random variables r with mean $\bar{r}$. The m_i servers of system S_i have identical execution speed s_i (measured by billion instructions per second (BIPS)). Hence, the task execution times on the servers of system S_i are i.i.d. exponential random variables $x_i = r/s_i$ with mean $\bar{x}_i = \bar{r}/s_i$.

Let $\mu_i = 1/\bar{x}_i = s_i/\bar{r}$ be the average service rate, i.e., the average number of tasks that can be finished by a server of S_i in one unit of time. The server utilization is

$$\rho_i = \frac{\lambda_i}{m_i\mu_i} = \frac{\lambda_i\bar{x}_i}{m_i} = \frac{\lambda_i\bar{r}}{m_i s_i},$$

which is the average percentage of time that a server of S_i is busy. Let $p_{i,k}$ denote the probability that there are k tasks (waiting or being processed) in the M/M/m queueing system for S_i. Then, we have ([24], p. 102)

$$p_{i,k} = \begin{cases} p_{i,0}\dfrac{(m_i\rho_i)^k}{k!}, & k \le m_i; \\[2ex] p_{i,0}\dfrac{m_i^{m_i}\rho_i^k}{m_i!}, & k \ge m_i; \end{cases}$$

where

$$p_{i,0} = \left(\sum_{k=0}^{m_i-1} \frac{(m_i\rho_i)^k}{k!} + \frac{(m_i\rho_i)^{m_i}}{m_i!} \cdot \frac{1}{1-\rho_i} \right)^{-1}.$$

The probability of queueing in S_i (i.e., the probability that a newly arrived task must wait because all servers are busy) is

$$P_{q,i} = \frac{p_{i,m_i}}{1-\rho_i} = p_{i,0}\frac{m_i^{m_i}}{m_i!} \cdot \frac{\rho_i^{m_i}}{1-\rho_i}.$$

The average number of tasks (in waiting or in execution) in S_i is

$$\bar{N}_i = \sum_{k=0}^{\infty} k p_{i,k} = m_i\rho_i + \frac{\rho_i}{1-\rho_i} P_{q,i}.$$

Applying Little's result ([24], p. 17), we get the average task response time of S_i as

$$T_i = \frac{\bar{N}_i}{\lambda_i} = \bar{x}_i + \frac{P_{q,i}}{m_i(1-\rho_i)}\bar{x}_i = \bar{x}_i \left(1 + \frac{P_{q,i}}{m_i(1-\rho_i)} \right).$$

In other words, the average task response time in multiserver system S_i is

$$T_i = \frac{\bar{r}}{s_i} \left(1 + p_{i,0}\frac{m_i^{m_i-1}}{m_i!} \cdot \frac{\rho_i^{m_i}}{(1-\rho_i)^2} \right).$$

3 Power Consumption

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 $P = aCV^2f$ (measured in Watt), where a is an activity factor, C is the loading capacitance, V is the supply voltage, and f is the clock frequency [9]. In the ideal case, the supply voltage and the clock frequency are related in such a way that $V \propto f^{\phi}$ for some constant $\phi > 0$ [45]. The processor execution speed s is usually linearly proportional to the clock frequency, namely, $s \propto f$. For ease of discussion, we will assume that $V = bf^{\phi}$ and $s = cf$, where b and c are some constants. Hence, we know that power consumption is $P = aCV^2f = ab^2Cf^{2\phi+1} = (ab^2C/c^{2\phi+1})s^{2\phi+1} = \xi s^{\alpha}$, where $\xi = ab^2C/c^{2\phi+1}$ and $\alpha = 2\phi+1$. For instance, by setting $b = 1.16$, $aC = 7.0$, $c = 1.0$, $\phi = 0.5$, $\alpha = 2\phi+1 = 2.0$, and $\xi = ab^2C/c^{\alpha} = 9.4192$, the value of P calculated by the equation $P = aCV^2f = \xi s^{\alpha}$ is reasonably close to that in [18] for the Intel Pentium M processor.

Since the multiserver systems considered in this paper are heterogeneous in the sense that each has its own ξ and α values, we assume that a server of S_i with speed s_i consumes power $\xi_i s_i^{\alpha_i}$. Notice that a server still consumes some amount of power even when it is idle. We assume that an idle server of S_i consumes certain base power P_i^*, which includes static power dissipation, short-circuit power dissipation, and other leakage and wasted power [2]. We will consider two types of server speed and power consumption models.

- In the *idle-speed model*, a server runs at zero speed when there is no task to perform. Since the power for speed s_i is $\xi_i s_i^{\alpha_i}$ and there are m_i servers, the average power consumption of multiserver system S_i is $P_i = m_i(\rho_i\xi_i s_i^{\alpha_i} + P_i^*) = \lambda_i\bar{r}\xi_i s_i^{\alpha_i-1} + m_iP_i^*$.
- In the *constant-speed model*, a server of S_i still runs at the speed s_i even if there is no task to perform. Hence, the power consumption of multiserver system S_i is $P_i = m_i(\xi_i s_i^{\alpha_i} + P_i^*)$.

4 Dynamic Speed and Power Management

The technique of *dynamic speed and power management* refers to dynamic server speed and power adjustment according to the current workload (i.e., the number of tasks in a multiserver system). Let the speed of the m_i servers of S_i be $s_{i,k}$ when there are k tasks in the queueing system, where $k \ge 0$. A sequence of server speeds $(s_{i,0}, s_{i,1}, s_{i,2}, s_{i,3}, ...)$ is called a *speed scheme* of S_i, which reflects and represents a strategy of workload dependent dynamic speed and power management. If $s_{i,1} = s_{i,2} = s_{i,3} = \cdots = s_i$, then we have a single-speed scheme for workload independent dynamic speed and power management, i.e., a standard M/M/m queueing system. Furthermore, if $s_{i,0} = 0$, we have the idle-speed mode; and if $s_{i,0} = s_i$, we have the constant-speed mode.

A multiserver system S_i with dynamic speed and power management can be characterized by a birth-death process ([24], p. 53). The states are 0, 1, 2, ..., k, ..., where state k means that there are k tasks in the multiserver system. The birth rate (i.e., the task arrival rate) is fixed at λ_i. The death rates (i.e., the task service rates) are $\mu_{i,k}$ with $k \ge 1$. Then, we have

$$\mu_{i,k} = \begin{cases} k\dfrac{s_{i,k}}{\bar{r}}, & 1 \le k \le m_i - 1; \\[2ex] m_i\dfrac{s_{i,k}}{\bar{r}}, & k \ge m_i. \end{cases}$$

This implies that ([24], p. 92)

$$\begin{aligned} p_{i,k} &= p_{i,0}\frac{\lambda_i^k}{\mu_{i,1}\mu_{i,2}\cdots\mu_{i,k}} \\ &= \begin{cases} p_{i,0}\dfrac{(\lambda_i\bar{r})^k}{k!}\cdot\dfrac{1}{s_{i,1}s_{i,2}\cdots s_{i,k}}, & 1 \le k \le m_i - 1; \\[2ex] p_{i,0}\dfrac{(\lambda_i\bar{r})^k}{m_i!m_i^{k-m_i}}\cdot\dfrac{1}{s_{i,1}s_{i,2}\cdots s_{i,k}}, & k \ge m_i; \end{cases} \end{aligned}$$

where

$$p_{i,0} = \left(1 + \sum_{k=1}^{m_i-1}\frac{(\lambda_i\bar{r})^k}{k!}\cdot\frac{1}{s_{i,1}s_{i,2}\cdots s_{i,k}} + \sum_{k=m_i}^{\infty}\frac{(\lambda_i\bar{r})^k}{m_i!m_i^{k-m_i}}\cdot\frac{1}{s_{i,1}s_{i,2}\cdots s_{i,k}}\right)^{-1}.$$

A speed scheme is valid if it results in a stable queueing system, i.e., $p_{i,0} > 0$.

Based on the $p_{i,k}$'s, we get the average number of tasks (in waiting or in execution) in S_i as

$$\bar{N}_i = \sum_{k=1}^{\infty} kp_{i,k}.$$

By Little's result, the average task response time of S_i is

$$T_i = \frac{\bar{N}_i}{\lambda_i}.$$

The average number of busy servers in S_i is

$$B_i = \sum_{k=0}^{m_i-1} kp_{i,k} + \sum_{k=m_i}^{\infty} m_i p_{i,k},$$

and the average server utilization of S_i is

$$\rho_i = \frac{B_i}{m_i}.$$

The average server speed of S_i is

$$\bar{s}_i = \sum_{k=0}^{\infty} p_{i,k}s_{i,k}.$$

The average power consumption of S_i is

$$\begin{aligned} P_i &= \sum_{k=0}^{m_i-1} p_{i,k}(k(\xi_i s_{i,k}^{\alpha_i} + P_i^*) + (m_i - k)P_i^*) \\ &\quad + \sum_{k=m_i}^{\infty} p_{i,k}m_i(\xi_i s_{i,k}^{\alpha_i} + P_i^*) \\ &= \xi_i\left(\sum_{k=0}^{m_i-1} p_{i,k}ks_{i,k}^{\alpha_i} + \sum_{k=m_i}^{\infty} p_{i,k}m_i s_{i,k}^{\alpha_i}\right) + m_iP_i^*, \end{aligned}$$

for the idle-speed model, and

$$P_i = \sum_{k=0}^{\infty} p_{i,k}m_i(\xi_i s_{i,k}^{\alpha_i} + P_i^*) = m_i\xi_i\sum_{k=0}^{\infty} p_{i,k}s_{i,k}^{\alpha_i} + m_iP_i^*,$$

for the constant-speed model.

5 d-SPEED SCHEMES

A d_i-speed scheme of S_i can be represented by $\psi_i = (b_{i,1}, b_{i,2}, ..., b_{i,d_i-1}; s_{i,1}, s_{i,2}, ..., s_{i,d_i})$, where $m_i < b_{i,1} < b_{i,2} < \cdots < b_{i,d_i-1}$, and $s_{i,1} < s_{i,2} < \cdots < s_{i,d_i}$. The speed of the m_i servers is $s_{i,1}$ when there are $k \le b_{i,1}$ tasks, and $s_{i,2}$ when there are $b_{i,1} + 1 \le k \le b_{i,2}$ tasks, ..., and s_{i,d_i-1} when there are $b_{i,d_i-2} + 1 \le k \le b_{i,d_i-1}$ tasks, and s_{i,d_i} when there are $k \ge b_{i,d_i-1} + 1$ tasks. Notice that the speed of an idle server is immaterial in this section. Therefore, we have

$$\mu_{i,k} = \begin{cases} k\dfrac{s_{i,1}}{\bar{r}}, & 1 \le k \le m_i - 1; \\[2ex] m_i\dfrac{s_{i,1}}{\bar{r}}, & m_i \le k \le b_{i,1}; \\[2ex] m_i\dfrac{s_{i,j}}{\bar{r}}, & b_{i,j-1} + 1 \le k \le b_{i,j},\ 2 \le j \le d_i - 1; \\[2ex] m_i\dfrac{s_{i,d_i}}{\bar{r}}, & k \ge b_{i,d_i-1} + 1. \end{cases}$$

This implies that

$$p_{i,k} = \begin{cases} p_{i,0}\dfrac{(\lambda_i\bar{r})^k}{k!}\cdot\dfrac{1}{s_{i,1}^k}, \\ \qquad 1 \le k \le m_i - 1; \\[2ex] p_{i,0}\dfrac{(\lambda_i\bar{r})^k}{m_i!m_i^{k-m_i}}\cdot\dfrac{1}{s_{i,1}^k}, \\ \qquad m_i \le k \le b_{i,1}; \\[2ex] p_{i,0}\dfrac{(\lambda_i\bar{r})^k}{m_i!m_i^{k-m_i}}\cdot\dfrac{1}{s_{i,1}^{b_{i,1}}s_{i,2}^{b_{i,2}-b_{i,1}}\cdots s_{i,j-1}^{b_{i,j-1}-b_{i,j-2}}s_{i,j}^{k-b_{i,j-1}}}, \\ \qquad b_{i,j-1} + 1 \le k \le b_{i,j},\ 2 \le j \le d_i - 1; \\[2ex] p_{i,0}\dfrac{(\lambda_i\bar{r})^k}{m_i!m_i^{k-m_i}}\cdot\dfrac{1}{s_{i,1}^{b_{i,1}}s_{i,2}^{b_{i,2}-b_{i,1}}\cdots s_{i,d_i-1}^{b_{i,d_i-1}-b_{i,d_i-2}}s_{i,d_i}^{k-b_{i,d_i-1}}}, \\ \qquad k \ge b_{i,d_i-1} + 1. \end{cases}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2706425, IEEE Transactions on Sustainable Computing

Let us define

$$\rho_{i,j} = \frac{\lambda_i \bar{r}}{m_i s_{i,j}},$$

which is the server utilization of S_i when its server speed is $s_{i,j}$, where $1 \le i \le n$, and $1 \le j \le d_i$. Then, we obtain

$$p_{i,k} = \begin{cases} p_{i,0}\dfrac{(m_i\rho_{i,1})^k}{k!}, \\ \qquad 1 \le k \le m_i - 1; \\ p_{i,0}\dfrac{m_i^{m_i}}{m_i!}\rho_{i,1}^k, \\ \qquad m_i \le k \le b_{i,1}; \\ p_{i,0}\dfrac{m_i^{m_i}}{m_i!}\rho_{i,1}^{b_{i,1}}\rho_{i,2}^{b_{i,2}-b_{i,1}}\cdots\rho_{i,j-1}^{b_{i,j-1}-b_{i,j-2}}\rho_{i,j}^{k-b_{i,j-1}}, \\ \qquad b_{i,j-1}+1 \le k \le b_{i,j},\ 2 \le j \le d_i - 1; \\ p_{i,0}\dfrac{m_i^{m_i}}{m_i!}\rho_{i,1}^{b_{i,1}}\rho_{i,2}^{b_{i,2}-b_{i,1}}\cdots\rho_{i,d_i-1}^{b_{i,d_i-1}-b_{i,d_i-2}}\rho_{i,d_i}^{k-b_{i,d_i-1}}, \\ \qquad k \ge b_{i,d_i-1}+1; \end{cases}$$

where

$$\begin{aligned} p_{i,0} \;=\; & \Bigg(1 + \sum_{k=1}^{m_i-1}\frac{(m_i\rho_{i,1})^k}{k!} + \sum_{k=m_i}^{b_{i,1}}\frac{m_i^{m_i}}{m_i!}\rho_{i,1}^k \\ & + \sum_{j=2}^{d_i-1}\sum_{k=b_{i,j-1}+1}^{b_{i,j}}\frac{m_i^{m_i}}{m_i!}\left(\prod_{l=1}^{j-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\rho_{i,j}^{k-b_{i,j-1}} \\ & + \sum_{k=b_{i,d_i-1}+1}^{\infty}\frac{m_i^{m_i}}{m_i!}\left(\prod_{l=1}^{d_i-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\rho_{i,d_i}^{k-b_{i,d_i-1}}\Bigg)^{-1}. \end{aligned}$$

In the above equation, we assume that $b_{i,0} = 0$ for all $1 \le i \le n$.

To continue the evaluation of $p_{i,0}$, we have

$$\begin{aligned} p_{i,0} \;=\; & \Bigg(1 + \sum_{k=1}^{m_i-1}\frac{(m_i\rho_{i,1})^k}{k!} + \frac{m_i^{m_i}}{m_i!}\sum_{k=m_i}^{b_{i,1}}\rho_{i,1}^k \\ & + \frac{m_i^{m_i}}{m_i!}\sum_{j=2}^{d_i-1}\left(\prod_{l=1}^{j-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\sum_{k=b_{i,j-1}+1}^{b_{i,j}}\rho_{i,j}^{k-b_{i,j-1}} \\ & + \frac{m_i^{m_i}}{m_i!}\left(\prod_{l=1}^{d_i-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\sum_{k=b_{i,d_i-1}+1}^{\infty}\rho_{i,d_i}^{k-b_{i,d_i-1}}\Bigg)^{-1} \\ =\; & \Bigg(1 + \sum_{k=1}^{m_i-1}\frac{(m_i\rho_{i,1})^k}{k!} + \frac{m_i^{m_i}}{m_i!}\cdot\frac{\rho_{i,1}^{m_i} - \rho_{i,1}^{b_{i,1}+1}}{1-\rho_{i,1}} \\ & + \frac{m_i^{m_i}}{m_i!}\sum_{j=2}^{d_i-1}\left(\prod_{l=1}^{j-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\frac{\rho_{i,j} - \rho_{i,j}^{b_{i,j}-b_{i,j-1}+1}}{1-\rho_{i,j}} \\ & + \frac{m_i^{m_i}}{m_i!}\left(\prod_{l=1}^{d_i-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\frac{\rho_{i,d_i}}{1-\rho_{i,d_i}}\Bigg)^{-1}. \end{aligned}$$

A speed scheme is valid if it results in a stable queueing system, i.e., $p_0 > 0$. It is clear that a d_i-speed scheme is valid if $\rho_{i,d_i} < 1$, i.e, $s_{i,d_i} > \lambda_i\bar{r}/m_i$.

In the following, we derive closed-form expressions of several major quantities of S_i, i.e., the average task response time T_i, the average server utilization ρ_i, the average server speed $\bar{s}_i$, and the average power consumption P_i. These closed-form expressions are critical to formulate and solve the optimization problems to be addressed in this paper.

Based on the $p_{i,k}$'s, we get

$$\begin{aligned} \bar{N}_i &= \sum_{k=1}^{\infty} k p_{i,k} \\ &= p_{i,0}\Bigg(\sum_{k=1}^{m_i-1}\frac{(m_i\rho_{i,1})^k}{(k-1)!} + \frac{m_i^{m_i}}{m_i!}\cdot \\ &\frac{m_i\rho_{i,1}^{m_i} - (m_i-1)\rho_{i,1}^{m_i+1} - (b_{i,1}+1)\rho_{i,1}^{b_{i,1}+1} + b_{i,1}\rho_{i,1}^{b_{i,1}+2}}{(1-\rho_{i,1})^2} \\ &+ \frac{m_i^{m_i}}{m_i!}\sum_{j=2}^{d_i-1}\left(\prod_{l=1}^{j-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right) \\ &\Big((b_{i,j-1}+1)\rho_{i,j} - b_{i,j-1}\rho_{i,j}^2 - (b_{i,j}+1)\rho_{i,j}^{b_{i,j}-b_{i,j-1}+1} \\ &+ b_{i,j}\rho_{i,j}^{b_{i,j}-b_{i,j-1}+2}\Big)\frac{1}{(1-\rho_{i,j})^2} \\ &+ \frac{m_i^{m_i}}{m_i!}\left(\prod_{l=1}^{d_i-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\frac{(b_{i,d_i-1}+1)\rho_{i,d_i} - b_{i,d_i-1}\rho_{i,d_i}^2}{(1-\rho_{i,d_i})^2}\Bigg). \end{aligned}$$

Hence, by using the above $\bar{N}_i$, the average task response time of S_i is

$$\begin{aligned} T_i &= \frac{\bar{N}_i}{\lambda_i} \\ &= \frac{p_{i,0}}{\lambda_i}\Bigg(\sum_{k=1}^{m_i-1}\frac{(m_i\rho_{i,1})^k}{(k-1)!} + \frac{m_i^{m_i}}{m_i!}\cdot \\ &\frac{m_i\rho_{i,1}^{m_i} - (m_i-1)\rho_{i,1}^{m_i+1} - (b_{i,1}+1)\rho_{i,1}^{b_{i,1}+1} + b_{i,1}\rho_{i,1}^{b_{i,1}+2}}{(1-\rho_{i,1})^2} \\ &+ \frac{m_i^{m_i}}{m_i!}\sum_{j=2}^{d_i-1}\left(\prod_{l=1}^{j-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right) \\ &\Big((b_{i,j-1}+1)\rho_{i,j} - b_{i,j-1}\rho_{i,j}^2 - (b_{i,j}+1)\rho_{i,j}^{b_{i,j}-b_{i,j-1}+1} \\ &+ b_{i,j}\rho_{i,j}^{b_{i,j}-b_{i,j-1}+2}\Big)\frac{1}{(1-\rho_{i,j})^2} \\ &+ \frac{m_i^{m_i}}{m_i!}\left(\prod_{l=1}^{d_i-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\frac{(b_{i,d_i-1}+1)\rho_{i,d_i} - b_{i,d_i-1}\rho_{i,d_i}^2}{(1-\rho_{i,d_i})^2}\Bigg), \end{aligned}$$

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

The average server utilization of S_i is $\rho_i = B_i/m_i$, where B_i is the average number of busy servers in S_i calculated by

$$\begin{aligned} B_i \;=\; & \sum_{k=1}^{m_i-1} k p_{i,k} + \sum_{k=m_i}^{\infty} m_i p_{i,k} \\ =\; & p_{i,0}\sum_{k=1}^{m_i-1}\frac{(m_i\rho_{i,1})^k}{(k-1)!} + m_i\left(1 - p_{i,0}\sum_{k=0}^{m_i-1}\frac{(m_i\rho_{i,1})^k}{k!}\right). \end{aligned}$$

The average server speed of S_i is

$$\begin{aligned}\bar{s}_i &= \sum_{k=0}^{m_i-1} p_{i,k}s_{i,1} + \sum_{k=m_i}^{b_{i,1}} p_{i,k}s_{i,1} \\ &+ \sum_{j=2}^{d_i-1}\sum_{k=b_{i,j-1}+1}^{b_{i,j}} p_{i,k}s_{i,j} + \sum_{k=b_{i,d_i-1}+1}^{\infty} p_{i,k}s_{i,d_i} \\ &= p_{i,0}\Bigg(\left(\sum_{k=0}^{m_i-1}\frac{(m_i\rho_{i,1})^k}{k!}\right)s_{i,1} + \frac{m_i^{m_i}}{m_i!}\cdot\frac{\rho_{i,1}^{m_i}-\rho_{i,1}^{b_{i,1}+1}}{1-\rho_{i,1}}s_{i,1} \\ &+ \frac{m_i^{m_i}}{m_i!}\left(\sum_{j=2}^{d_i-1}\left(\prod_{l=1}^{j-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\frac{\rho_{i,j}-\rho_{i,j}^{b_{i,j}-b_{i,j-1}+1}}{1-\rho_{i,j}}\right)s_{i,j} \\ &+ \frac{m_i^{m_i}}{m_i!}\left(\left(\prod_{l=1}^{d_i-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\frac{\rho_{i,d_i}}{1-\rho_{i,d_i}}\right)s_{i,d_i}\Bigg).\end{aligned}$$

Assume that the speed of an idle server is $s_{i,0}$. For the idle-speed model, we have $s_{i,0} = 0$. For the constant-speed model, we have $s_{i,0} = s_{i,1}$. The average power consumption by the m_i servers in S_i is

$$\begin{aligned}P_i &= \sum_{k=0}^{m_i-1} p_{i,k}(k(\xi_i s_{i,1}^{\alpha_i} + P_i^*) + (m_i-k)(\xi_i s_{i,0}^{\alpha_i} + P_i^*)) \\ &+ \sum_{k=m_i}^{b_{i,1}} p_{i,k}m_i(\xi_i s_{i,1}^{\alpha_i} + P_i^*) \\ &+ \sum_{j=2}^{d_i-1}\sum_{k=b_{i,j-1}+1}^{b_{i,j}} p_{i,k}m_i(\xi_i s_{i,j}^{\alpha_i} + P_i^*) \\ &+ \sum_{k=b_{i,d_i-1}+1}^{\infty} p_{i,k}m_i(\xi_i s_{i,d_i}^{\alpha_i} + P_i^*) \\ &= \xi_i p_{i,0}\Bigg(\left(\sum_{k=0}^{m_i-1}(m_i-k)\frac{(m_i\rho_{i,1})^k}{k!}\right)s_{i,0}^{\alpha_i} \\ &+ \left(\sum_{k=1}^{m_i-1}\frac{(m_i\rho_{i,1})^k}{(k-1)!} + \frac{m_i^{m_i+1}}{m_i!}\cdot\frac{\rho_{i,1}^{m_i}-\rho_{i,1}^{b_{i,1}+1}}{1-\rho_{i,1}}\right)s_{i,1}^{\alpha_i} \\ &+ \frac{m_i^{m_i+1}}{m_i!}\sum_{j=2}^{d_i-1}\Bigg(\left(\prod_{l=1}^{j-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right) \\ &\quad \frac{\rho_{i,j}-\rho_{i,j}^{b_{i,j}-b_{i,j-1}+1}}{1-\rho_{i,j}}\Bigg)s_{i,j}^{\alpha_i} \\ &+ \frac{m_i^{m_i+1}}{m_i!}\left(\left(\prod_{l=1}^{d_i-1}\rho_{i,l}^{b_{i,l}-b_{i,l-1}}\right)\frac{\rho_{i,d_i}}{1-\rho_{i,d_i}}\right)s_{i,d_i}^{\alpha_i}\Bigg) + m_iP_i^*.\end{aligned}$$

The average task response time T of a group of n heterogeneous multiserver systems $S_1, S_2, ..., S_n$ is

$$T(\lambda_1,\lambda_2,...,\lambda_n) = \frac{\lambda_1}{\lambda}T_1 + \frac{\lambda_2}{\lambda}T_2 + \cdots + \frac{\lambda_n}{\lambda}T_n,$$

where T is treated as a function of load distribution $\lambda_1, \lambda_2, ..., \lambda_n$.

The average server utilization ρ of a group of n heterogeneous multiserver systems $S_1, S_2, ..., S_n$ is

$$\rho(\lambda_1,\lambda_2,...,\lambda_n) = \frac{\lambda_1}{\lambda}\rho_1 + \frac{\lambda_2}{\lambda}\rho_2 + \cdots + \frac{\lambda_n}{\lambda}\rho_n,$$

where ρ is treated as a function of load distribution $\lambda_1, \lambda_2, ..., \lambda_n$.

The average server speed $\bar{s}$ of a group of n heterogeneous multiserver systems $S_1, S_2, ..., S_n$ is

$$\bar{s}(\lambda_1,\lambda_2,...,\lambda_n) = \frac{\lambda_1}{\lambda}\bar{s}_1 + \frac{\lambda_2}{\lambda}\bar{s} + \cdots + \frac{\lambda_n}{\lambda}\bar{s}_n,$$

where $\bar{s}$ is treated as a function of load distribution $\lambda_1, \lambda_2, ..., \lambda_n$.

The average power consumption P of a group of n heterogeneous multiserver systems $S_1, S_2, ..., S_n$ is

$$P(\lambda_1,\lambda_2,...,\lambda_n) = \frac{\lambda_1}{\lambda}P_1 + \frac{\lambda_2}{\lambda}P_2 + \cdots + \frac{\lambda_n}{\lambda}P_n,$$

where P is treated as a function of load distribution $\lambda_1, \lambda_2, ..., \lambda_n$.

6 Minimizing Average Task Response Time

In this section, we formulate and solve our optimal task dispatching problem with minimized average task response time for multiple heterogeneous multiserver systems with dynamic d-speed and power management.

6.1 Problem Definition

Our *optimal task dispatching problem with minimized average task response time* for multiple heterogeneous multiserver systems with dynamic d-speed and power management can be specified as follows: given the number n of multiserver systems, the sizes of the multiserver systems $m_1, m_2, ..., m_n$, a d_i-speed scheme $\psi_i = (b_{i,1}, b_{i,2}, ..., b_{i,d_i-1}, s_{i,1}, s_{i,2}, ..., s_{i,d_i})$ of S_i, for all $1 \le i \le n$, the power consumption model parameters $\xi_1, \alpha_1, \xi_2, \alpha_2, ..., \xi_n, \alpha_n$, the base power consumption $P_1^*, P_2^*, ..., P_n^*$, the average task execution requirement $\bar{r}$, and the task arrival rate λ, find a load distribution, i.e., the task arrival rates $\lambda_1, \lambda_2, ..., \lambda_n$ to the multiserver systems, such that the average task response time $T(\lambda_1, \lambda_2, ..., \lambda_n)$ is minimized, subject to the constraint

$$F(\lambda_1,\lambda_2,...,\lambda_n) = \lambda,$$

where

$$F(\lambda_1,\lambda_2,...,\lambda_n) = \lambda_1 + \lambda_2 + \cdots + \lambda_n,$$

and $\rho_i < 1$, for all $1 \le i \le n$.

6.2 An Algorithm

The above optimization problem can be solved by using the method of Lagrange multiplier, i.e.,

$$\nabla T(\lambda_1,\lambda_2,...,\lambda_n) = \phi\nabla F(\lambda_1,\lambda_2,...,\lambda_n),$$

that is,

$$\frac{\partial T}{\partial \lambda_i} = \phi\frac{\partial F}{\partial \lambda_i} = \phi,$$

for all $1 \le i \le n$, where ϕ is a Lagrange multiplier.

As we see below, $\partial T/\partial\lambda_i$ is an extremely complicated function of λ_i. Hence, an analytical solution is virtually impossible to find. Instead, an algorithm for finding numerical values of $\lambda_1, \lambda_2, ..., \lambda_n$ and ϕ can be developed. The algorithm works as follows. We notice that $\partial T/\partial\lambda_i$ is an increasing function of λ_i. Therefore, given a ϕ, we can find λ_i, $1 \le i \le n$, by the bisection algorithm. The obtained $\lambda_1, \lambda_2, ..., \lambda_n$ are used to verify the condition $F(\lambda_1, \lambda_2, ..., \lambda_n) = \lambda$, and such verification can be employed to find ϕ, again by the bisection method.

In the following, we give $\partial T/\partial\lambda_i$. Notice that $\partial\rho_{i,j}/\partial\lambda_i = \bar{r}/m_i s_{i,j}$, for all $1 \le i \le n$, and $1 \le j \le d_i$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2706425, IEEE Transactions on Sustainable Computing

Hence, we have

$$\frac{\partial T}{\partial \lambda_i} = \frac{1}{\lambda}\left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i}\right),$$

where

$$\begin{aligned}
\frac{\partial T_i}{\partial \lambda_i} &= -\frac{T_i}{\lambda_i} + \frac{T_i}{p_{i,0}} \cdot \frac{\partial p_{i,0}}{\partial \lambda_i} \\
&+ \frac{p_{i,0}}{\lambda_i}\Bigg(\sum_{k=1}^{m_i-1} \frac{m_i^k}{(k-1)!} k \rho_{i,1}^{k-1} \frac{\bar{r}}{m_i s_{i,1}} \\
&+ \frac{m_i^{m_i}}{m_i!}\Bigg(\Big(m_i^2 \rho_{i,1}^{m_i-1} - (m_i^2-1)\rho_{i,1}^{m_i} - (b_{i,1}+1)^2 \rho_{i,1}^{b_{i,1}} \\
&\qquad + b_{i,1}(b_{i,1}+2)\rho_{i,1}^{b_{i,1}+1} \Big) \frac{1}{(1-\rho_{i,1})^2} \\
&\quad + 2\Big(m_i \rho_{i,1}^{m_i} - (m_i-1)\rho_{i,1}^{m_i+1} - (b_{i,1}+1)\rho_{i,1}^{b_{i,1}+1} \\
&\qquad + b_{i,1}\rho_{i,1}^{b_{i,1}+2} \Big) \frac{1}{(1-\rho_{i,1})^3} \Bigg) \frac{\bar{r}}{m_i s_{i,1}} \\
&+ \frac{m_i^{m_i}}{m_i!} \sum_{j=2}^{d_i-1} \Bigg(\Bigg(\sum_{l=1}^{j-1} \Bigg(\prod_{l' \neq l} \rho_{i,l'}^{b_{i,l'}-b_{i,l'-1}} \Bigg) \\
&\qquad (b_{i,l}-b_{i,l-1}) \rho_{i,l}^{b_{i,l}-b_{i,l-1}-1} \frac{\bar{r}}{m_i s_{i,l}} \Bigg) \\
&\times \Big((b_{i,j-1}+1)\rho_{i,j} - b_{i,j-1}\rho_{i,j}^2 - (b_{i,j}+1)\rho_{i,j}^{b_{i,j}-b_{i,j-1}+1} \\
&\quad + b_{i,j}\rho_{i,j}^{b_{i,j}-b_{i,j-1}+2} \Big) \frac{1}{(1-\rho_{i,j})^2} \\
&+ \Bigg(\prod_{l=1}^{j-1} \rho_{i,l}^{b_{i,l}-b_{i,l-1}} \Bigg) \\
&\times \Bigg(\frac{1}{(1-\rho_{i,j})^2} \Big((b_{i,j-1}+1) - 2 b_{i,j-1}\rho_{i,j} \\
&\qquad - (b_{i,j}+1)(b_{i,j}-b_{i,j-1}+1)\rho_{i,j}^{b_{i,j}-b_{i,j-1}} \\
&\qquad + b_{i,j}(b_{i,j}-b_{i,j-1}+2)\rho_{i,j}^{b_{i,j}-b_{i,j-1}+1} \Big) \\
&+ 2\Big((b_{i,j-1}+1)\rho_{i,j} - b_{i,j-1}\rho_{i,j}^2 - (b_{i,j}+1)\rho_{i,j}^{b_{i,j}-b_{i,j-1}+1} \\
&\quad + b_{i,j}\rho_{i,j}^{b_{i,j}-b_{i,j-1}+2} \Big) \frac{1}{(1-\rho_{i,j})^3} \Bigg) \frac{\bar{r}}{m_i s_{i,j}} \Bigg) \\
&+ \frac{m_i^{m_i}}{m_i!} \Bigg(\Bigg(\sum_{l=1}^{d_i-1} \Bigg(\prod_{l' \neq l} \rho_{i,l'}^{b_{i,l'}-b_{i,l'-1}} \Bigg) \\
&\qquad (b_{i,l}-b_{i,l-1}) \rho_{i,l}^{b_{i,l}-b_{i,l-1}-1} \frac{\bar{r}}{m_i s_{i,l}} \Bigg) \\
&\qquad \times \Bigg(\frac{(b_{i,d_i-1}+1)\rho_{i,d_i} - b_{i,d_i-1}\rho_{i,d_i}^2}{(1-\rho_{i,d_i})^2} \Bigg) \\
&\quad + \Bigg(\prod_{l=1}^{d_i-1} \rho_{i,l}^{b_{i,l}-b_{i,l-1}} \Bigg) \\
&\qquad \times \Bigg(\frac{(b_{i,d_i-1}+1) - 2 b_{i,d_i-1}\rho_{i,d_i}}{(1-\rho_{i,d_i})^2} \\
&\qquad + \frac{2\left((b_{i,d_i-1}+1)\rho_{i,d_i} - b_{i,d_i-1}\rho_{i,d_i}^2 \right)}{(1-\rho_{i,d_i})^3} \Bigg) \frac{\bar{r}}{m_i s_{i,d_i}} \Bigg) \Bigg),
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial p_{i,0}}{\partial \lambda_i} &= -p_{i,0}^2 \Bigg(\sum_{k=1}^{m_i-1} \frac{m_i^k}{k!} k \rho_{i,1}^{k-1} \frac{\bar{r}}{m_i s_{i,1}} \\
&+ \frac{m_i^{m_i}}{m_i!} \Bigg(\frac{m_i \rho_{i,1}^{m_i-1} - (b_{i,1}+1)\rho_{i,1}^{b_{i,1}}}{1-\rho_{i,1}} + \frac{\rho_{i,1}^{m_i} - \rho_{i,1}^{b_{i,1}+1}}{(1-\rho_{i,1})^2} \Bigg) \frac{\bar{r}}{m_i s_{i,1}} \\
&+ \frac{m_i^{m_i}}{m_i!} \sum_{j=2}^{d_i-1} \Bigg(\Bigg(\sum_{l=1}^{j-1} \Bigg(\prod_{l' \neq l} \rho_{i,l'}^{b_{i,l'}-b_{i,l'-1}} \Bigg) \\
&\qquad (b_{i,l}-b_{i,l-1}) \rho_{i,l}^{b_{i,l}-b_{i,l-1}-1} \frac{\bar{r}}{m_i s_{i,l}} \Bigg) \\
&\qquad \times \Bigg(\frac{\rho_{i,j} - \rho_{i,j}^{b_{i,j}-b_{i,j-1}+1}}{1-\rho_{i,j}} \Bigg) \\
&\quad + \Bigg(\prod_{l=1}^{j-1} \rho_{i,l}^{b_{i,l}-b_{i,l-1}} \Bigg) \\
&\qquad \times \Bigg(\frac{1 - (b_{i,j}-b_{i,j-1}+1)\rho_{i,j}^{b_{i,j}-b_{i,j-1}}}{1-\rho_{i,j}} \\
&\qquad\quad + \frac{\rho_{i,j} - \rho_{i,j}^{b_{i,j}-b_{i,j-1}+1}}{(1-\rho_{i,j})^2} \Bigg) \frac{\bar{r}}{m_i s_{i,j}} \Bigg) \\
&+ \frac{m_i^{m_i}}{m_i!} \Bigg(\Bigg(\sum_{l=1}^{d_i-1} \Bigg(\prod_{l' \neq l} \rho_{i,l'}^{b_{i,l'}-b_{i,l'-1}} \Bigg) \\
&\qquad (b_{i,l}-b_{i,l-1}) \rho_{i,l}^{b_{i,l}-b_{i,l-1}-1} \frac{\bar{r}}{m_i s_{i,l}} \Bigg) \Bigg(\frac{\rho_{i,d_i}}{1-\rho_{i,d_i}} \Bigg) \\
&\quad + \Bigg(\prod_{l=1}^{d_i-1} \rho_{i,l}^{b_{i,l}-b_{i,l-1}} \Bigg) \\
&\qquad \Bigg(\frac{1}{1-\rho_{i,d_i}} + \frac{\rho_{i,d_i}}{(1-\rho_{i,d_i})^2} \Bigg) \frac{\bar{r}}{m_i s_{i,d_i}} \Bigg) \Bigg),
\end{aligned}$$

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

6.3 Numerical Data

Let us consider $n = 7$ heterogeneous multiserver systems $S_1, S_2, ..., S_7$. The sizes are $m_1 = 3$, $m_2 = 3$, $m_3 = 5$, $m_4 = 5$, $m_5 = 5$, $m_6 = 7$, $m_7 = 7$, respectively. The values of d_i are $d_1 = 2$, $d_2 = 3$, $d_3 = 2$, $d_4 = 3$, $d_5 = 4$, $d_6 = 2$, $d_7 = 3$, respectively. The d-speed schemes are $\psi_1 = (5; 1.0, 1.5)$, $\psi_2 = (6, 9; 1.0, 1.3, 1.6)$, $\psi_3 = (6; 1.0, 1.4)$, $\psi_4 = (8, 12; 1.0, 1.3, 1.6)$, $\psi_5 = (10, 15, 20; 1.0, 1.2, 1.5, 1.8)$, $\psi_6 = (11; 1.0, 1.3)$, $\psi_7 = (14, 21; 1.0, 1.2, 1.4)$, respectively. S_1 and S_2 have the same size but different d-speed schemes. S_3, S_4, and S_5 have the same size but different d-speed schemes. S_6 and S_7 have the same size but different d-speed schemes. We set $\xi_i = 2.0$, $\alpha_i = 3.0$, and $P_i^* = 2.0$, for all $1 \le i \le n$. Also, we set $\bar{r} = 1$. It is clear that the maximum task arrival rate is

$$\lambda_{\max} = \sum_{i=1}^{n} \frac{m_i s_{i,d_i}}{\bar{r}}.$$

In our example, we have $\lambda_{\max} = 52.2$.

In Table 1, we show the optimal load distribution $\lambda_1, \lambda_2, ..., \lambda_7$ which gives the minimized average task response

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2706425, IEEE Transactions on Sustainable Computing

TABLE 1
Example of Optimal Load Distribution for Minimized Response Time

λ	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
2.6100000	0.0473071	0.0472956	0.3197633	0.3097695	0.3097233	0.7881092	0.7880320
7.8300000	0.3154758	0.3123869	1.1821506	1.0370448	1.0314151	1.9796175	1.9719093
13.0500000	0.7009647	0.6720685	2.2696756	1.7582472	1.7162086	2.9944324	2.9384029
18.2700000	1.1919042	1.0766962	3.4140124	2.4999040	2.3538749	3.9623531	3.7712554
23.4900000	1.8445035	1.5320024	4.2758345	3.3450729	2.9698706	4.9916418	4.5310743
28.7100000	2.5364377	2.0498104	4.8672405	4.4113331	3.5715823	6.0374878	5.2361082
33.9300000	3.0375862	2.7406338	5.3106865	5.6730410	4.2829612	6.8832824	6.0018089
39.1500000	3.3337301	3.3514801	5.6109929	6.2974411	6.3145647	7.3729934	6.8687977
44.3700000	3.6742732	3.8851955	5.9913695	6.8582394	7.7243267	7.9032366	8.3333592
49.5900000	4.2088832	4.4975042	6.6376575	7.6105212	8.5855438	8.6851887	9.3647014

time for $\lambda = (2j-1)\lambda_{\text{step}}$, where $\lambda_{\text{step}} = \lambda_{\text{max}}/20$ and $j = 1, 2, 3, ..., 10$. We observe that when λ is small to moderate, S_1 is allocated more load than S_2, since S_1 is more sensitive to the increased workload and increases its speed earlier than S_2. Furthermore, the increased speed is higher than that of S_2. However, as λ becomes large, S_2 is allocated more load than S_1, because the ultimate speed of S_2 is higher than that of S_1. Similar situation also exists in the group of S_3, S_4, S_5. When λ is small, S_3 receives more load than S_4, and S_4 receives more load than S_5. As λ increases, S_3 receives less load than S_4, but S_4 still receives more load than S_5. As λ further increases, S_3 receives less load than S_4, and S_4 receives less load than S_5. Similar situation also exists in the group of S_6 and S_7. When λ is small to moderate, S_6 is assigned more load than S_7. However, as λ becomes large, S_6 is assigned less load than S_7.

6.4 Performance Comparison

We compare the performance (i.e., the average task response time) of a group of heterogeneous multiserver systems with dynamic speed and power management with that of the same group of heterogeneous multiserver systems with static speed and power management. In particular, we turn each multiserver system S_i with a d_i-speed scheme into a system with a 1-speed scheme of speed s_i. The speed s_i is determined in such a way that the power consumption of S_i is still P_i. Hence, we have

$$s_i = \left(\frac{P_i - m_i P_i^*}{\lambda_i \bar{r} \xi_i}\right)^{1/(\alpha_i - 1)},$$

for the idle-speed model, and

$$s_i = \left(\frac{1}{\xi_i}\left(\frac{P_i}{m_i} - P_i^*\right)\right)^{1/\alpha_i},$$

for the constant-speed model.

Consider the same group of heterogeneous multiserver systems specified in Section 6.3. For $\lambda = x\lambda_{\text{max}}$, where $x = 0.55, 0.65, 0.75, 0.85, 0.95$, we show in Tables 2 and 3 (for the idle-speed model and the constant-speed model respectively): (1) the optimal load distribution $\lambda_1, \lambda_2, ..., \lambda_n$; (2) ρ_i, $\bar{s}_i$, T_i, and $\rho(\lambda_1, \lambda_2, ..., \lambda_n)$, $\bar{s}(\lambda_1, \lambda_2, ..., \lambda_n)$, $T(\lambda_1, \lambda_2, ..., \lambda_n)$ with dynamic speed and power management; (3) ρ_i, s_i, T_i and $\rho(\lambda_1, \lambda_2, ..., \lambda_n)$, $s(\lambda_1, \lambda_2, ..., \lambda_n)$, $T(\lambda_1, \lambda_2, ..., \lambda_n)$ with static speed and power management; (4) P_i and $P(\lambda_1, \lambda_2, ..., \lambda_n)$. It is observed that for the same P_i, the server S_i with dynamic speed and power management has higher average server utilization, slower average server speed, and shorter average task response time than the server S_i with static speed and power management. The difference is more noticeable when the server utilization gets higher.

7 Minimizing Average Power Consumption

In this section, we formulate and solve our optimal task dispatching problem with minimized average power consumption for multiple heterogeneous multiserver systems with dynamic d-speed and power management.

7.1 Problem Definition

Our *optimal task dispatching problem with minimized average power consumption* for multiple heterogeneous multiserver systems with dynamic d-speed and power management can be specified as follows: given the number n of multiserver systems, the sizes of the multiserver systems $m_1, m_2, ..., m_n$, a d_i-speed scheme $\psi_i = (b_{i,1}, b_{i,2}, ..., b_{i,d_i-1}, s_{i,1}, s_{i,2}, ..., s_{i,d_i})$ of S_i, for all $1 \le i \le n$, the power consumption model parameters $\xi_1, \alpha_1, \xi_2, \alpha_2, ..., \xi_n, \alpha_n$, the base power consumption $P_1^*, P_2^*, ..., P_n^*$, the average task execution requirement $\bar{r}$, and the task arrival rate λ, find a load distribution, i.e., the task arrival rates $\lambda_1, \lambda_2, ..., \lambda_n$ to the multiserver systems, such that the average power consumption $P(\lambda_1, \lambda_2, ..., \lambda_n)$ is minimized, subject to the constraint

$$F(\lambda_1, \lambda_2, ..., \lambda_n) = \lambda,$$

where

$$F(\lambda_1, \lambda_2, ..., \lambda_n) = \lambda_1 + \lambda_2 + \cdots + \lambda_n,$$

and $\rho_i < 1$, for all $1 \le i \le n$.

7.2 An Algorithm

The above optimization problem can be solved by using the method of Lagrange multiplier, i.e.,

$$\nabla P(\lambda_1, \lambda_2, ..., \lambda_n) = \phi \nabla F(\lambda_1, \lambda_2, ..., \lambda_n),$$

that is,

$$\frac{\partial P}{\partial \lambda_i} = \phi \frac{\partial F}{\partial \lambda_i} = \phi,$$

for all $1 \le i \le n$, where ϕ is a Lagrange multiplier.

As we see below, $\partial P/\partial \lambda_i$ is an extremely complicated function of λ_i. Hence, an analytical solution is virtually impossible to find. Instead, an algorithm for finding numerical values of $\lambda_1, \lambda_2, ..., \lambda_n$ and ϕ can be developed. The algorithm works as follows. We notice that $\partial P/\partial \lambda_i$ is an increasing function of λ_i. Therefore, given a ϕ, we can find λ_i, $1 \le i \le n$, by the bisection algorithm. The obtained $\lambda_1, \lambda_2, ..., \lambda_n$ are used to verify the condition $F(\lambda_1, \lambda_2, ..., \lambda_n) = \lambda$, and such verification can be employed to find ϕ, again by the bisection method.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2706425, IEEE Transactions on Sustainable Computing

TABLE 2
Numerical Data for Response Time Comparison (Idle-Speed Model)

i	λ_i	Dynamic Management			Static Management			P_i
		ρ_i	$\bar{s}_i$	T_i	ρ_i	s_i	T_i	
				$\lambda = 28.71$				
1	2.5364377	0.7681015	1.0773778	1.3206634	0.7295129	1.1589641	1.4331542	12.8138752
2	2.0498104	0.6658566	1.0174135	1.2972947	0.6569453	1.0400715	1.3639821	10.4347598
3	4.8672405	0.8436211	1.1298270	1.1814020	0.8089300	1.2033774	1.3280669	24.0966683
4	4.4113331	0.8254002	1.0568664	1.2763990	0.8018232	1.1003256	1.4202736	20.6817466
5	3.5715823	0.7076388	1.0066777	1.2117941	0.7052512	1.0128540	1.2460723	17.3279807
6	6.0374878	0.8270231	1.0354752	1.2119036	0.8139019	1.0597079	1.3171497	27.5599671
7	5.2361082	0.7436926	1.0043228	1.1737509	0.7422977	1.0077027	1.1975529	24.6341673
	Average	0.7828258	1.0479060	1.2253758	0.7659659	1.0823589	1.3177838	21.5840358
				$\lambda = 33.93$				
1	3.0375862	0.8608391	1.1516896	1.5009202	0.8102051	1.2497190	1.7331485	15.4881891
2	2.7406338	0.8354326	1.0781120	1.5921013	0.8033635	1.1371498	1.8523446	13.0878804
3	5.3106865	0.8871277	1.1750096	1.2852018	0.8521317	1.2464474	1.5172944	26.5016955
4	5.6730410	0.9439792	1.1906290	1.5315626	0.8970483	1.2648240	1.9516355	28.1512313
5	4.2829612	0.8285640	1.0280283	1.4200655	0.8183943	1.0466742	1.5763157	19.3841986
6	6.8832824	0.9032609	1.0800652	1.3598377	0.8818255	1.1151028	1.6273449	31.1180939
7	6.0018089	0.8402473	1.0171540	1.3373252	0.8349485	1.0268912	1.4477434	26.6578806
	Average	0.8776920	1.1019695	1.4118793	0.8506748	1.1502849	1.6537761	24.7737396
				$\lambda = 39.15$				
1	3.3337301	0.9030598	1.2081835	1.6710562	0.8516498	1.3048126	2.0289610	17.3515898
2	3.3514801	0.9292783	1.1878817	1.9237247	0.8791795	1.2706847	2.4817211	16.8228657
3	5.6109929	0.9129110	1.2092876	1.3997695	0.8798831	1.2753952	1.7216034	28.2540501
4	6.2974411	0.9729608	1.2865274	1.7157636	0.9259867	1.3601580	2.3673518	33.3009065
5	6.3145647	0.9882542	1.2746587	2.1621042	0.9369008	1.3479687	2.7332767	32.9473751
6	7.3729934	0.9365007	1.1167841	1.4960937	0.9127924	1.1539149	1.9360035	33.6345694
7	6.8687977	0.9249884	1.0562684	1.6080587	0.9103614	1.0778761	2.0320806	29.9605692
	Average	0.9418463	1.1860614	1.6961956	0.9055762	1.2453045	2.1747421	29.2285994
				$\lambda = 44.37$				
1	3.6742732	0.9414832	1.2832745	2.0150094	0.8962696	1.3665059	2.6377040	19.7222231
2	3.8851955	0.9728436	1.3222216	2.3854605	0.9253194	1.3995872	3.4642047	21.2209868
3	5.9913695	0.9416697	1.2566042	1.6540466	0.9138010	1.3113073	2.1668005	30.6046396
4	6.8582394	0.9878085	1.3838393	2.0114442	0.9494535	1.4446709	3.0928130	38.6273082
5	7.7243267	0.9992725	1.5455929	2.6593743	0.9616366	1.6064960	3.5613942	49.8703370
6	7.9032366	0.9639673	1.1650665	1.7557030	0.9417963	1.1988089	2.5368038	36.7161578
7	8.3333592	0.9897380	1.2007419	2.2478005	0.9657327	1.2327219	3.8517882	39.3268016
	Average	0.9745432	1.3077382	2.1078654	0.9419364	1.3608135	3.0876894	36.2025506
				$\lambda = 49.59$				
1	4.2088832	0.9839852	1.4189758	4.2143967	0.9635830	1.4559835	6.5458335	23.8447225
2	4.4975042	0.9951155	1.5040526	4.6078904	0.9744646	1.5384532	8.7289062	27.2897289
3	6.6376575	0.9817239	1.3458075	3.3930260	0.9694290	1.3693953	5.1468832	34.8944486
4	7.6105212	0.9976013	1.5245030	3.6918481	0.9819158	1.5501373	7.4587024	46.5750338
5	8.5855438	0.9999345	1.7171742	4.2490716	0.9857502	1.7419309	8.3453191	62.1026290
6	8.6851887	0.9908721	1.2498691	3.3237435	0.9802346	1.2657595	6.1642549	41.8298999
7	9.3647014	0.9990413	1.3387731	3.9472290	0.9885942	1.3532493	9.6783993	48.2988507
	Average	0.9935923	1.4399575	3.8595099	0.9796433	1.4631126	7.5329490	43.5160866

In the following, we give $\partial P/\partial \lambda_i$. It is clear that

$$\frac{\partial P}{\partial \lambda_i} = \frac{1}{\lambda}\left(P_i + \lambda_i \frac{\partial P_i}{\partial \lambda_i} \right),$$

where

$$\begin{aligned}
\frac{\partial P_i}{\partial \lambda_i} &= \frac{P_i - m_i P_i^*}{p_{i,0}} \cdot \frac{\partial p_{i,0}}{\partial \lambda_i} \\
&+\xi_i p_{i,0} \Bigg(\Bigg(\sum_{k=0}^{m_i-1} (m_i - k) \frac{m_i^k}{k!} k \rho_{i,1}^{k-1} \frac{\bar{r}}{m_i s_{i,1}} \Bigg) s_{i,0}^{\alpha_i} \\
&+ \Bigg(\sum_{k=1}^{m_i-1} \frac{m_i^k}{(k-1)!} k \rho_{i,1}^{k-1} \frac{\bar{r}}{m_i s_{i,1}} \\
&\qquad + \frac{m_i^{m_i+1}}{m_i!} \Bigg(\frac{m \rho_{i,1}^{m_i-1} - (b_{i,1}+1) \rho_{i,1}^{b_{i,1}}}{1-\rho_{i,1}} \\
&\qquad\qquad + \frac{\rho_{i,1}^{m_i} - \rho_{i,1}^{b_{i,1}+1}}{(1-\rho_{i,1})^2} \Bigg) \frac{\bar{r}}{m_i s_{i,1}} \Bigg) s_{i,1}^{\alpha_i} \\
&+ \frac{m_i^{m_i+1}}{m_i!} \sum_{j=2}^{d_i-1} \Bigg(\Bigg(\sum_{l=1}^{j-1} \Bigg(\prod_{l' \neq l} \rho_{i,l'}^{b_{i,l'}-b_{i,l'-1}} \Bigg. \Bigg. \Bigg. \\
&\qquad (b_{i,l} - b_{i,l-1}) \rho_{i,l}^{b_{i,l}-b_{i,l-1}-1} \frac{\bar{r}}{m_i s_{i,l}} \Bigg) \frac{\rho_{i,j} - \rho_{i,j}^{b_{i,j}-b_{i,j-1}+1}}{1-\rho_{i,j}} \\
&\quad + \Bigg(\prod_{l=1}^{j-1} \rho_{i,l}^{b_{i,l}-b_{i,l-1}} \Bigg) \\
&\qquad \Bigg(\frac{1 - (b_{i,j} - b_{i,j-1} + 1) \rho_{i,j}^{b_{i,j}-b_{i,j-1}}}{1-\rho_{i,j}} \\
&\qquad\quad + \frac{\rho_{i,j} - \rho_{i,j}^{b_{i,j}-b_{i,j-1}+1}}{(1-\rho_{i,j})^2} \Bigg) \frac{\bar{r}}{m_i s_{i,j}} \Bigg) s_{i,j}^{\alpha_i} \\
&+ \frac{m_i^{m_i+1}}{m_i!} \Bigg(\Bigg(\sum_{l=1}^{d_i-1} \Bigg(\prod_{l' \neq l} \rho_{i,l'}^{b_{i,l'}-b_{i,l'-1}} \Bigg. \Bigg. \Bigg. \\
&\qquad (b_{i,l} - b_{i,l-1}) \rho_{i,l}^{b_{i,l}-b_{i,l-1}-1} \frac{\bar{r}}{m_i s_{i,l}} \Bigg) \frac{\rho_{i,d_i}}{1-\rho_{i,d_i}} \\
&\quad + \Bigg(\prod_{l=1}^{d_i-1} \rho_{i,l}^{b_{i,l}-b_{i,l-1}} \Bigg)
\end{aligned}$$

TABLE 3
Numerical Data for Response Time Comparison (Constant-Speed Model)

i	λ_i	Dynamic Management			Static Management			
		ρ_i	$\bar{s}_i$	T_i	ρ_i	s_i	T_i	P_i
				$\lambda = 28.71$				
1	2.5364377	0.7681015	1.0773778	1.3206634	0.7617091	1.1099766	1.6394386	14.2052663
2	2.0498104	0.6658566	1.0174135	1.2972947	0.6673538	1.0238500	1.4126404	12.4396201
3	4.8672405	0.8436211	1.1298270	1.1814020	0.8382585	1.1612743	1.5321694	25.6604574
4	4.4113331	0.8254002	1.0568664	1.2763990	0.8206080	1.0751377	1.5466112	22.4277449
5	3.5715823	0.7076388	1.0066777	1.2117941	0.7084245	1.0083171	1.2579981	20.2515928
6	6.0374878	0.8270231	1.0354752	1.2119036	0.8252656	1.0451159	1.3796808	29.9816443
7	5.2361082	0.7436926	1.0043228	1.1737509	0.7440947	1.0052691	1.2037111	28.2224708
	Average	0.7828258	1.0479060	1.2253758	0.7805240	1.0617881	1.4092527	23.9108833
				$\lambda = 33.93$				
1	3.0375862	0.8608391	1.1516896	1.5009202	0.8449940	1.1982674	2.1300620	16.3231546
2	2.7406338	0.8354326	1.0781120	1.5921013	0.8274229	1.1040842	2.1169668	14.0752849
3	5.3106865	0.8871277	1.1750096	1.2852018	0.8792110	1.2080573	1.8100569	27.6304184
4	5.6730410	0.9439792	1.1906290	1.5315626	0.9207538	1.2322601	2.4694299	28.7114392
5	4.2829612	0.8285640	1.0280283	1.4200655	0.8273424	1.0353540	1.6461111	21.0985588
6	6.8832824	0.9032609	1.0800652	1.3598377	0.8965313	1.0968117	1.8068107	32.4724419
7	6.0018089	0.8402473	1.0171540	1.3373252	0.8398833	1.0208576	1.4808309	28.8944185
	Average	0.8776920	1.1019695	1.4118793	0.8689203	1.1253476	1.8941521	26.0853500
				$\lambda = 39.15$				
1	3.3337301	0.9030598	1.2081835	1.6710562	0.8836364	1.2575799	2.5908590	17.9332308
2	3.3514801	0.9292783	1.1878817	1.9237247	0.9060477	1.2330036	3.1928939	17.2471958
3	5.6109929	0.9129110	1.2092876	1.3997695	0.9040720	1.2412713	2.1016687	29.1249406
4	6.2974411	0.9729608	1.2865274	1.7157636	0.9463816	1.3308461	3.1876247	33.5712982
5	6.3145647	0.9882542	1.2746587	2.1621042	0.9558505	1.3212452	3.8143407	33.0648328
6	7.3729934	0.9365007	1.1167841	1.4960937	0.9271947	1.1359910	2.2477931	34.5235600
7	6.8687977	0.9249884	1.0562684	1.6080587	0.9195691	1.0670833	2.2208682	31.0107316
	Average	0.9418463	1.1860614	1.6961956	0.9247316	1.2188718	2.7360927	29.8533778
				$\lambda = 44.37$				
1	3.6742732	0.9414832	1.2832745	2.0150094	0.9217965	1.3286639	3.4983297	20.0733238
2	3.8851955	0.9728436	1.3222216	2.3854605	0.9462073	1.3686907	4.8056462	21.3839252
3	5.9913695	0.9416697	1.2566042	1.6540466	0.9329531	1.2843881	2.7241209	31.1879425
4	6.8582394	0.9878085	1.3838393	2.0114442	0.9646444	1.4219207	4.3350344	38.7492228
5	7.7243267	0.9992725	1.5455929	2.6593743	0.9741988	1.5857804	5.2064454	49.8776121
6	7.9032366	0.9639673	1.1650665	1.7557030	0.9538023	1.1837189	3.1048496	37.2206160
7	8.3333592	0.9897380	1.2007419	2.2478005	0.9751822	1.2207769	5.1873599	39.4704693
	Average	0.9745432	1.3077382	2.1078654	0.9569137	1.3392853	4.1820913	36.4616056
				$\lambda = 49.59$				
1	4.2088832	0.9839852	1.4189758	4.2143967	0.9738274	1.4406670	9.1009019	23.9408112
2	4.4975042	0.9951155	1.5040526	4.6078904	0.9824525	1.5259446	12.6936196	27.3190359
3	6.6376575	0.9817239	1.3458075	3.3930260	0.9771287	1.3586046	6.8073596	35.0772092
4	7.6105212	0.9976013	1.5245030	3.6918481	0.9876913	1.5410728	10.8689528	46.5990211
5	8.5855438	0.9999345	1.7171742	4.2490716	0.9904733	1.7336245	12.3983244	62.1032839
6	8.6851887	0.9908721	1.2498691	3.3237435	0.9852735	1.2592861	8.1585711	41.9576906
7	9.3647014	0.9990413	1.3387731	3.9472290	0.9922522	1.3482605	14.0993900	48.3122719
	Average	0.9935923	1.4399575	3.8595099	0.9855452	1.4543288	10.7408595	43.5800730

$$\left(\frac{1}{1-\rho_{i,d_i}} + \frac{\rho_{i,d_i}}{(1-\rho_{i,d_i})^2}\right)\frac{\bar{r}}{m_i s_{i,d_i}}\Bigg) s_{i,d_i}^{\alpha_i}\Bigg),$$

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

7.3 Numerical Data

Consider the same group of heterogeneous multiserver systems specified in Section 6.3. In Tables 4 and 5, for the idle-speed model and the constant-speed model respectively, we show the optimal load distribution $\lambda_1, \lambda_2, ..., \lambda_7$ which gives the minimized average power consumption for $\lambda = (2j-1)\lambda_{\text{step}}$, where $\lambda_{\text{step}} = \lambda_{\text{max}}/20$ and $j = 1, 2, 3, ..., 10$. We observe that optimal task dispatching with minimized average power consumption is trickier than optimal task dispatching with minimized average task response time due to situations of underflow and overflow. Let us consider

$$\beta_i = P_i + \lambda_i \frac{\partial P_i}{\partial \lambda_i},$$

where $1 \le i \le n$. It is required that $\beta_i = \lambda\phi$ for all $1 \le i \le n$. It is clear that $\beta_i \ge P_i$, and $P_i \ge m_i P_i^*$ for the idle-speed model and $P_i \ge m_i(\xi_i s_{i,1}^{\alpha_i} + P_i^*)$ for the constant-speed model. Hence, if λ is too small, the condition $\beta_i = \lambda\phi$ may not be satisfied by some multiserver system S_i. In this case, we have to set $\lambda_i = 0$, which implies that $P_i = m_i P_i^*$ (for the idle-speed model) or $P_i = m_i(\xi_i s_{i,1}^{\alpha_i} + P_i^*)$ (for the constant-speed model), $\beta_i = P_i$ (which is greater than $\lambda\phi$, i.e., *underflow*), and $\rho_i = 0$. For instances, in Table 4, the above situation happens to S_6 and S_7 when $\lambda = 2.61$. In Table 5, the above situation happens to S_3, S_4, S_5, S_6, S_7 when $\lambda = 2.61$, and to S_6 and S_7 when $\lambda = 7.83$ and $\lambda = 13.05$. Furthermore, it is observed that $\partial P_i/\partial \lambda_i$ approaches an upper bound π_i as λ_i increases. For example, for S_1, $\partial P_1/\partial \lambda_1$ approaches $\pi_1 = 8.4208957$ for the idle-speed model and $\pi_1 = 8.1331346$ for the constant-speed model. Hence, if λ is too large, the condition $\beta_i = \lambda\phi$ may not be satisfied by some multiserver system S_i. In this case, we can only set λ_i sufficiently close to $m_i s_{i,b_i}/\bar{r}$, and ρ_i is sufficiently close to 1, and the β_i is sufficiently close to $m_i(\xi_i s_{i,b_i}^{\alpha_i} + P_i^*) + (m_i s_{i,b_i}/\bar{r})\pi_i$ (which is less than $\lambda\phi$, i.e., *overflow*, and for S_1, this value is 64.1440306 for the idle-speed model and 62.8491057 for the constant-speed model). For instances, in Table 4, the above situation happens to S_1 when $\lambda = 39.15$, and to S_1, S_2, S_3 when $\lambda = 44.37$, and to S_1, S_2, S_3, S_6 when $\lambda = 49.59$. In Table 5, the above situation happens to S_1, S_2, S_3 when $\lambda = 44.37$, and to S_1, S_2, S_3, S_6 when $\lambda = 49.59$.

TABLE 4
Example of Optimal Load Distribution for Minimized Power Consumption (Idle-Speed Model)

λ	Parameter	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$
2.61	λ_i	1.0842747	1.1195798	0.1353818	0.1353818	0.1353818	0.0000000	0.0000000
	P_i	8.2016087	8.2478319	10.2707637	10.2707637	10.2707637	14.0000000	14.0000000
	β_i	10.5415274	10.5415274	10.5415274	10.5415274	10.5415274	14.0000000	14.0000000
	ρ_i	0.3599556	0.3727164	0.0270764	0.0270764	0.0270764	0.0000000	0.0000000
7.83	λ_i	1.7480736	1.8865460	1.2323621	1.2403749	1.2408643	0.2408896	0.2408896
	P_i	9.8457235	9.9808234	12.4696113	12.4809785	12.4817377	14.4817792	14.4817792
	β_i	14.9635583	14.9635583	14.9635583	14.9635583	14.9635583	14.9635583	14.9635583
	ρ_i	0.5671545	0.6179113	0.2463270	0.2480673	0.2481725	0.0344128	0.0344128
13.05	λ_i	2.1041572	2.2502850	2.0378964	2.1417901	2.1683407	1.1737649	1.1737657
	P_i	11.0116666	11.0665374	14.1613166	14.2989045	14.3387714	16.3475301	16.3475314
	β_i	18.6950629	18.6950629	18.6950629	18.6950629	18.6950629	18.6950629	18.6950629
	ρ_i	0.6656812	0.7212069	0.4050339	0.4278490	0.4335893	0.1676807	0.1676808
18.27	λ_i	2.4242222	2.5444642	2.7136032	2.9606360	3.1164610	2.2549374	2.2556761
	P_i	12.3017405	12.1862428	15.7973290	16.0729244	16.2897739	18.5101494	18.5113583
	β_i	23.0227898	23.0227898	23.0227898	23.0227898	23.0227898	23.0227898	23.0227898
	ρ_i	0.7434831	0.7937213	0.5317051	0.5871705	0.6211933	0.3221274	0.3222393
23.49	λ_i	2.7568488	2.8273641	3.3197318	3.6017301	3.8324120	3.5527627	3.5991505
	P_i	13.9127109	13.5307130	17.5853866	17.7523412	17.9985596	21.1269397	21.2005105
	β_i	28.4254980	28.4254980	28.4254980	28.4254980	28.4254980	28.4254980	28.4254980
	ρ_i	0.8123268	0.8520676	0.6357939	0.7028162	0.7546236	0.5070260	0.5141046
28.71	λ_i	3.1545249	3.1449687	3.9550678	4.1649502	4.3698283	4.8213631	5.0992969
	P_i	16.2003006	15.3982259	19.8983294	19.6634171	19.6963828	23.9294509	24.3204646
	β_i	35.8363871	35.8363871	35.8363871	35.8363871	35.8363871	35.8363871	35.8363871
	ρ_i	0.8785638	0.9034851	0.7318411	0.7915979	0.8414630	0.6819165	0.7252111
33.93	λ_i	3.7405568	3.5872684	4.7829476	4.7950995	4.9122087	5.9257910	6.1861280
	P_i	20.2088845	18.6452106	23.6655155	22.5358645	22.0979863	27.1615742	27.2467587
	β_i	47.8873399	47.8873399	47.8873399	47.8873399	47.8873399	47.8873399	47.8873399
	ρ_i	0.9478402	0.9521980	0.8345770	0.8711373	0.9096010	0.8152470	0.8609434
39.15	λ_i	4.4999995	4.1450298	5.7531812	5.4501998	5.4487614	6.8972309	6.9555974
	P_i	26.2499962	23.6842132	29.1162547	26.5402368	25.4177574	31.1851463	30.3765193
	β_i	64.1440234	64.4091020	64.4091020	64.4091020	64.4091020	64.4091020	64.4091020
	ρ_i	1.0000000	0.9849413	0.9241513	0.9294415	0.9544152	0.9043181	0.9315286
44.37	λ_i	4.4999995	4.7999995	6.9999993	6.2302999	6.0743116	8.0142542	7.7511359
	P_i	26.2499962	30.5759947	37.4399950	32.7049145	30.6002548	37.4040910	35.0237865
	β_i	64.1440234	83.8936667	87.4831403	87.5626383	87.5626383	87.5626383	87.5626383
	ρ_i	1.0000000	1.0000000	1.0000000	0.9705634	0.9823964	0.9686970	0.9744389
49.59	λ_i	4.4999995	4.7999995	6.9999993	7.6943158	7.2059347	9.0999991	9.2897521
	P_i	26.2499962	30.5759947	37.4399950	47.5052151	43.1079740	44.7579934	47.6051582
	β_i	64.1440234	83.8936667	87.4831403	133.2705387	133.2705387	110.3166738	133.2705387
	ρ_i	1.0000000	1.0000000	1.0000000	0.9982302	0.9977916	1.0000000	0.9987734

7.4 Performance Comparison

We compare the cost (i.e., the average power consumption) of a group of heterogeneous multiserver systems with dynamic speed and power management with that of the same group of heterogeneous multiserver systems with static speed and power management. In particular, we turn each multiserver system S_i with a d_i-speed scheme into a system with a 1-speed scheme of speed s_i. The speed s_i is determined in such a way that the average task response time of S_i is still T_i.

Consider the same group of heterogeneous multiserver systems specified in Section 6.3. For $\lambda = x\lambda_{\text{max}}$, where $x = 0.35, 0.45, 0.55, 0.65$, (which are chosen such that both underflow and overflow do not happen to any server,) we show in Tables 6 and 7 (for the idle-speed model and the constant-speed model respectively): (1) the optimal load distribution $\lambda_1, \lambda_2, ..., \lambda_n$; (2) ρ_i, $\bar{s}_i$, P_i, and $\rho(\lambda_1, \lambda_2, ..., \lambda_n)$, $\bar{s}(\lambda_1, \lambda_2, ..., \lambda_n)$, $P(\lambda_1, \lambda_2, ..., \lambda_n)$ with dynamic speed and power management; (3) ρ_i, s_i, P_i and $\rho(\lambda_1, \lambda_2, ..., \lambda_n)$, $s(\lambda_1, \lambda_2, ..., \lambda_n)$, $P(\lambda_1, \lambda_2, ..., \lambda_n)$ with static speed and power management; (4) T_i and $T(\lambda_1, \lambda_2, ..., \lambda_n)$. It is observed that for the same T_i, the server S_i with dynamic speed and power management has higher average server utilization, slower average server speed, and less average power consumption than the server S_i with static speed and power management. The difference is more noticeable when the server utilization gets higher.

8 Minimizing Average Cost-Performance Ratio

For a multiserver system S_i, our performance measure is $1/T_i$, which is inversely proportional to the average task response time T_i, the higher, the better. There are many different factors which determine the cost of cloud computing. Since the number of servers m_i is fixed in dynamic speed and power management, our cost measure is mainly the cost of power consumption P_i, the lower, the better. The cost-performance (or price-performance) ratio (CPR) refers to a product's ability to deliver performance for its cost. Generally speaking, products with a lower CPR are more desirable, excluding other factors. In this paper, we define CPR as cost/performance, i.e., $R_i = P_iT_i$. The average cost-performance ratio R of a group of n heterogeneous multiserver systems S_1, S_2, ..., S_n is

$$R(\lambda_1, \lambda_2, ..., \lambda_n) = \frac{\lambda_1}{\lambda}R_1 + \frac{\lambda_2}{\lambda}R_2 + \cdots + \frac{\lambda_n}{\lambda}R_n$$
$$= \frac{\lambda_1}{\lambda}P_1T_1 + \frac{\lambda_2}{\lambda}P_2T_2 + \cdots + \frac{\lambda_n}{\lambda}P_nT_n,$$

where R is treated as a function of load distribution $\lambda_1, \lambda_2, ..., \lambda_n$.

In this section, we formulate and solve our optimal task dispatching problem with minimized average cost-performance ratio for multiple heterogeneous multiserver systems with dynamic d-speed and power management.

(Due to space limitation, the remaining content of this section is moved to the supplementary file.)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2706425, IEEE Transactions on Sustainable Computing

TABLE 5
Example of Optimal Load Distribution for Minimized Power Consumption (Constant-Speed Model)

λ	Parameter	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$
2.61	λ_i	1.1728906	1.4371094	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
	P_i	12.0627557	12.0541020	20.0000000	20.0000000	20.0000000	28.0000000	28.0000000
	β_i	12.3837221	12.3837221	20.0000000	20.0000000	20.0000000	28.0000000	28.0000000
	ρ_i	0.3887616	0.4768231	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
7.83	λ_i	2.3281863	2.4942605	0.5951178	1.0014405	1.4109949	0.0000000	0.0000000
	P_i	13.5607284	13.2835920	20.0000662	20.0000540	20.0000455	28.0000000	28.0000000
	β_i	20.0004961	20.0004961	20.0004961	20.0004961	20.0004961	28.0000000	28.0000000
	ρ_i	0.7212997	0.7821605	0.1190220	0.2002868	0.2821977	0.0000000	0.0000000
13.05	λ_i	2.3996846	2.5532548	2.1584024	2.7422196	3.1964385	0.0000000	0.0000000
	P_i	13.7665044	13.4495537	20.1505621	20.1191198	20.0971320	28.0000000	28.0000000
	β_i	20.9431494	20.9431494	20.9431494	20.9431494	20.9431494	28.0000000	28.0000000
	ρ_i	0.7379123	0.7957092	0.4282272	0.5454971	0.6366746	0.0000000	0.0000000
18.27	λ_i	2.8465108	2.9098180	3.4739741	3.8717132	4.1768601	0.3628308	0.6777372
	P_i	15.4222361	14.7769559	21.4982894	21.1318170	20.9034529	28.0000000	28.0000000
	β_i	28.0000000	28.0000000	28.0000000	28.0000000	28.0000000	28.0000000	28.0000000
	ρ_i	0.8287585	0.8668426	0.6604304	0.7472092	0.8121575	0.0518330	0.0968196
23.49	λ_i	2.8479667	2.9109495	3.4768436	3.8739398	4.1787035	2.7299974	3.4715995
	P_i	15.4286726	14.7821008	21.5036659	21.1357722	20.9065892	28.0024245	28.0019663
	β_i	28.0259031	28.0259031	28.0259031	28.0259031	28.0259031	28.0259031	28.0259031
	ρ_i	0.8290179	0.8670383	0.6608810	0.7475620	0.8124483	0.3899562	0.4959042
28.71	λ_i	3.0382954	3.0574255	3.8273903	4.1395930	4.3968461	4.8480393	5.4024105
	P_i	16.3267062	15.4996066	22.2568723	21.6862564	21.3433191	28.3996312	28.3094296
	β_i	31.5452797	31.5452797	31.5452797	31.5452797	31.5452797	31.5452797	31.5452797
	ρ_i	0.8609509	0.8907800	0.7137149	0.7879319	0.8453720	0.6854229	0.7657734
33.93	λ_i	3.6198098	3.4905742	4.7276233	4.7726405	4.9062870	6.0592611	6.3538041
	P_i	19.7121798	18.2119946	25.1021726	23.7307516	22.9771479	30.0296034	29.5291440
	β_i	43.4882453	43.4882453	43.4882453	43.4882453	43.4882453	43.4882453	43.4882453
	ρ_i	0.9360005	0.9435795	0.8285024	0.8686990	0.9089808	0.8292750	0.8786217
39.15	λ_i	4.4088190	4.0575231	5.7585727	5.4359878	5.4333716	7.0119530	7.0437729
	P_i	25.5138652	22.9459282	29.9036852	27.1574566	25.7737714	32.9683220	31.6881024
	β_i	60.8379086	60.8379086	60.8379086	60.8379086	60.8379086	60.8379086	60.8379086
	ρ_i	0.9954356	0.9813833	0.9245658	0.9284289	0.9534291	0.9127652	0.9377844
44.37	λ_i	4.4999995	4.7999995	6.9999993	6.1990138	6.0391802	8.0643531	7.7674546
	P_i	26.2499963	30.5759947	37.4399953	32.7365625	30.4593159	38.1288777	35.4843477
	β_i	62.8490985	83.5509420	84.2102690	84.5471817	84.5471817	84.5471817	84.5471817
	ρ_i	1.0000000	1.0000000	1.0000000	0.9693935	0.9813557	0.9707243	0.9750250
49.59	λ_i	4.4999995	4.7999995	6.9999993	7.6968120	7.2023225	9.0999991	9.2908680
	P_i	26.2499963	30.5759947	37.4399953	47.5505645	43.0850376	44.7579937	47.6325623
	β_i	62.8490985	83.5509420	84.2102690	132.8200740	132.8200740	107.9557001	132.8200740
	ρ_i	1.0000000	1.0000000	1.0000000	0.9982479	0.9977754	1.0000000	0.9987777

9 RELATED RESEARCH

There have been extensive research in cloud load balancing and load distribution. Several surveys and comparative studies have been conducted. In [4], existing load balancing techniques in cloud computing were discussed and compared based on various parameters. In [16], the authors explored autonomic approaches for optimizing provisioning for heterogeneous workloads on enterprise grids and clouds, and reviewed load balancing strategies for cloud infrastructures. In [20], the author surveyed various dynamic load balancing algorithms in cloud with discussion and comparison of the pros and cons of these algorithms. In [21], the authors presented a comparative study of various load balancing schemes in different cloud environments based on requirements specified in service level agreement. In [22], the authors gave an overview of many load balancing algorithms which help to achieve better throughput and improve the response time in cloud environments. In [23], the authors gave an overview of load balancing in cloud computing by exposing the most important research challenges. In [30], the authors investigated the different algorithms proposed to resolve the issue of load balancing and task scheduling in cloud computing, and discussed and compared these algorithms to provide an overview of the latest approaches in the field. In [33], the authors provided a comprehensive review on the existing load balancing strategies and presented load balancer as a service model adopted by the major market players. In [35], the authors presented a survey of dynamic load balancing strategies on cloud, with focus on various metrics to analyze the efficacy of existing techniques. In [37], various load balancing algorithms were compared on the basis of their metrics.

Numerous researchers have investigated various approaches to cloud load balancing. In [5], the authors showed a new approach to dynamic load balancing using the concept of mobile agent, i.e., a software program which executes independently and performs the basic task. In [10], the authors proposed a novel load balancing strategy using a genetic algorithm, which thrives to balance the load of a cloud infrastructure while trying to minimize the makespan of a given task set. In [11], the authors proposed an algorithm named honey bee behavior inspired load balancing, which aims to achieve well balanced load across virtual machines for maximizing the throughput and minimizing the amount of waiting time of the tasks. In [12], the authors proposed a novel approach to dynamic load balancing in cloud computing systems based on the phenomena of self-organization in a game theoretical spatially generalized prisoner's dilemma model defined on the two-dimensional cellular automata space. In [14], the authors focused on two load balancing algorithms in cloud, i.e., Min-Min and Max-Min, to minimize response time and waiting time. In [15], the authors used an agent-based dynamic load balancing approach which greatly reduces the communication cost of servers, accelerates the rate of load balancing, and improves the throughput and response time of the cloud. In [27], the author studied the problem of optimal distribution of generic tasks over

TABLE 6
Numerical Data for Power Consumption Comparison (Idle-Speed Model)

i	λ_i	Dynamic Management			Static Management			T_i
		ρ_i	$\bar{s}_i$	P_i	ρ_i	s_i	P_i	
				$\lambda = 18.27$				
1	2.4242222	0.7434831	1.0645909	12.3017405	0.6894232	1.1721016	12.6609003	1.2898882
2	2.5444642	0.7937213	1.0544334	12.1862428	0.7416644	1.1435829	12.6552086	1.5005392
3	2.7136032	0.5317051	1.0110156	15.7973290	0.5220176	1.0396596	15.8662241	1.0224552
4	2.9606360	0.5871705	1.0049567	16.0729244	0.5798343	1.0212008	16.1750044	1.0780187
5	3.1164610	0.6211933	1.0020989	16.2897739	0.6168970	1.0103667	16.3628216	1.1228339
6	2.2549374	0.3221274	1.0000066	18.5101494	0.3221195	1.0000447	18.5102779	1.0018714
7	2.2556761	0.3222393	1.0000002	18.5113583	0.3222390	1.0000014	18.5113652	1.0019188
	Average	0.5688214	1.0189496	15.6291556	0.5510368	1.0539329	15.7813768	1.1455737
				$\lambda = 23.49$				
1	2.7568488	0.8123268	1.1066228	13.9127109	0.7424444	1.2377352	14.4469210	1.3894011
2	2.8273641	0.8520676	1.0903871	13.5307130	0.7846219	1.2011577	14.1585278	1.6344130
3	3.3197318	0.6357939	1.0281525	17.5853866	0.6125646	1.0838797	17.8000096	1.0426273
4	3.6017301	0.7028162	1.0175298	17.7523412	0.6807017	1.0582404	18.0669587	1.1507922
5	3.8324120	0.7546236	1.0118588	17.9985596	0.7359993	1.0414173	18.3128848	1.2784317
6	3.5527627	0.5070260	1.0005116	21.1269397	0.5062434	1.0025564	21.1419006	1.0206776
7	3.5991505	0.5141046	1.0000597	21.2005105	0.5139661	1.0003857	21.2038553	1.0248849
	Average	0.6740868	1.0320808	17.8488773	0.6479153	1.0801009	18.1197701	1.2035740
				$\lambda = 28.71$				
1	3.1545249	0.8785638	1.1729445	16.2003006	0.7972748	1.3188782	16.9742114	1.5595601
2	3.1449687	0.9034851	1.1448378	15.3982259	0.8232977	1.2733217	16.1981788	1.8009095
3	3.9550678	0.7318411	1.0591724	19.8983294	0.6904233	1.1456935	20.3829522	1.0773785
4	4.1649502	0.7915979	1.0413921	19.6634171	0.7504242	1.1100255	20.2637421	1.2347995
5	4.3698283	0.8414630	1.0325027	19.6963828	0.8045418	1.0862899	20.3130199	1.4507935
6	4.8213631	0.6819165	1.0068496	23.9294509	0.6718277	1.0252124	24.1350892	1.0847631
7	5.0992969	0.7252111	1.0032599	24.3204646	0.7184896	1.0138921	24.4839225	1.1520848
	Average	0.7825571	1.0557009	20.3966297	0.7446551	1.1208445	20.8805643	1.3037979
				$\lambda = 33.93$				
1	3.7405568	0.9478402	1.2990120	20.2088845	0.8731989	1.4279133	21.2535143	2.1182163
2	3.5872684	0.9521980	1.2435581	18.6452106	0.8660691	1.3806705	19.6764684	2.0904415
3	4.7829476	0.8345770	1.1220125	23.6655155	0.7731772	1.2372190	24.6426200	1.1671458
4	4.7950995	0.8711373	1.0878826	22.5358645	0.8086086	1.1860126	23.4898211	1.3461104
5	4.9122087	0.9096010	1.0728407	22.0979863	0.8540097	1.1503871	23.0015395	1.6586429
6	5.9257910	0.8152470	1.0312946	27.1615742	0.7823460	1.0820552	27.8763473	1.1969086
7	6.1861280	0.8609434	1.0227892	27.2467587	0.8322456	1.0618651	27.9504300	1.3877706
	Average	0.8769587	1.1084997	23.6305970	0.8231054	1.1945310	24.5112835	1.5114816

a group of heterogeneous blade servers in a cloud computing environment or a data center, such that the average response time of generic tasks is minimized. In [34], the authors introduced a threshold based dynamic compare and balance algorithm for cloud server optimization, which also minimizes the number of host machines to be powered on for reducing the cost of cloud services. In [38], the authors proposed an autonomous agent-based load balancing algorithm, which provides dynamic load balancing for cloud environment. In [39], an enhanced shortest job first scheduling algorithm was used to achieve reduced response time and reduced starvation and job rejection rate. In [41], the authors developed an approach from machine learning to learn task arrival and execution patterns online, i.e., automatically acquiring such knowledge without any beforehand modeling and proactively allocating tasks on account of the forthcoming tasks and their execution dynamics. In [42], the authors studied the collaboration among benevolent clouds that are cooperative in nature and willing to accept jobs from other clouds, and took advantage of machine learning, and proposed a distributed scheduling mechanism to learn the knowledge of job model, resource performance, and others' policies. In [43], the authors proposed a fairness-aware load balancing algorithm, where the load balancing problem is defined as a game, and the Nash equilibrium solution for this problem minimizes the expected response time, while maintaining fairness.

Cloud load distribution has been considered together with energy consumption. In [6], the authors conducted a survey of research in energy-efficient computing and proposed architectural principles for energy-efficient management of clouds and energy-efficient resource allocation policies and scheduling algorithms considering QoS expectations and power usage characteristics of the devices. In [8], the authors addressed optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers as optimization problems. In [13], the authors proposed a new power-aware load balancing algorithm based on artificial bee colony to detect both over-utilized and under-utilized hosts for effective power management. In [17], the authors studied the problem of power consumption minimization with performance constraint in heterogeneous distributed embedded systems by optimal load distribution. In [19], the authors discussed existing load balancing techniques in cloud computing and further compared them based on various parameters and discussed these techniques from energy consumption and carbon emission perspective. In [26], the author considered 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. In [29], the authors modeled a data center as a cyber physical system to capture the thermal properties exhibited by the data center, where software aspects such as scheduling, load balancing, and computations are the cyber component, and hardware aspects such as servers and switches are the physical component. In [32], the authors investigated load distribution strategies to minimize electricity cost and increase renewable energy integration subject to compliance with service level agreement, with consideration of the adverse effects of switching the servers. In [40], the authors

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2706425, IEEE Transactions on Sustainable Computing

TABLE 7
Numerical Data for Power Consumption Comparison (Constant-Speed Model)

		Dynamic Management			Static Management			
i	λ_i	ρ_i	$\bar{s}_i$	P_i	ρ_i	s_i	P_i	T_i
				$\lambda = 18.27$				
1	2.8465108	0.8287585	1.1200785	15.4222361	0.7554397	1.2560062	17.8884866	1.4214638
2	2.9098180	0.8668426	1.1030967	14.7769559	0.7955205	1.2192513	16.8750411	1.6757537
3	3.4739741	0.6604304	1.0343644	21.4982894	0.6329626	1.0976869	23.2262120	1.0494653
4	3.8717132	0.7472092	1.0271335	21.1318170	0.7163783	1.0809131	22.6290978	1.1889633
5	4.1768601	0.8121575	1.0232146	20.9034529	0.7822343	1.0679307	22.1794919	1.3839079
6	0.3628308	0.0518330	1.0000000	28.0000000	0.0518330	1.0000000	28.0000000	1.0000000
7	0.6777372	0.0968196	1.0000000	28.0000000	0.0968196	1.0000000	28.0000000	1.0000012
	Average	0.7414020	1.0554264	19.6959689	0.7000220	1.1287641	21.3519564	1.3132156
				$\lambda = 23.49$				
1	2.8479667	0.8290179	1.1203043	15.4286726	0.7556470	1.2563038	17.8969394	1.4220083
2	2.9109495	0.8670383	1.1032782	14.7821008	0.7956655	1.2195031	16.8817807	1.6763290
3	3.4768436	0.6608810	1.0344878	21.5036659	0.6333324	1.0979522	23.2358027	1.0496007
4	3.8739398	0.7475620	1.0272260	21.1357722	0.7166550	1.0811170	22.6362473	1.1892949
5	4.1787035	0.8124483	1.0232924	20.9065892	0.7824597	1.0680943	22.1850897	1.3845224
6	2.7299974	0.3899562	1.0000434	28.0024245	0.3898984	1.0002596	28.0109066	1.0053327
7	3.4715995	0.4959042	1.0000386	28.0019663	0.4958160	1.0002557	28.0107423	1.0205400
	Average	0.6922031	1.0411335	21.4829495	0.6599336	1.0983336	22.7759601	1.2455967
				$\lambda = 28.71$				
1	3.0382954	0.8609509	1.1518143	16.3267062	0.7819008	1.2952604	19.0383468	1.5012496
2	3.0574255	0.8907800	1.1283619	15.4996066	0.8134792	1.2528185	17.7981997	1.7528783
3	3.8273903	0.7137149	1.0517631	22.2568723	0.6760116	1.1323445	24.5189688	1.0688158
4	4.1395930	0.7879319	1.0399867	21.6862564	0.7476681	1.1073344	23.5780185	1.2306709
5	4.3968461	0.8453720	1.0339972	21.3433191	0.8074595	1.0890567	22.9166967	1.4605413
6	4.8480393	0.6854229	1.0071542	28.3996312	0.6749374	1.0261352	29.1266143	1.0867920
7	5.4024105	0.7657734	1.0059996	28.3094296	0.7546193	1.0227315	28.9765892	1.2030461
	Average	0.7840352	1.0499455	22.8637143	0.7469300	1.1136188	24.4590540	1.2990498
				$\lambda = 33.93$				
1	3.6198098	0.9360005	1.2706028	19.7121798	0.8572006	1.4076089	22.7339050	1.9419825
2	3.4905742	0.9435795	1.2199452	18.2119946	0.8574522	1.3569558	20.9916131	2.0174694
3	4.7276233	0.8285024	1.1170223	25.1021726	0.7680658	1.2310465	28.6562063	1.1584887
4	4.7726405	0.8686990	1.0858291	23.7307516	0.8068443	1.1830388	26.5575855	1.3418731
5	4.9062870	0.9089808	1.0722767	22.9771479	0.8535666	1.1495968	25.1927595	1.6562885
6	6.0592611	0.8292750	1.0363338	30.0296034	0.7931842	1.0913086	32.1957843	1.2149213
7	6.3538041	0.8786217	1.0290646	29.5291440	0.8455852	1.0734415	31.3166234	1.4372288
	Average	0.8786241	1.1022567	25.0270766	0.8235910	1.1898375	27.5701743	1.4904961

investigated performance and power tradeoff for multiple heterogeneous servers by considering two problems, i.e., optimal job scheduling with fixed service rates and joint optimal service speed scaling and job scheduling. In [44], the authors employed a game theoretic approach to solving the problem of minimizing energy consumption as a Stackelberg game, and modeled the problem of minimizing average task response time as a noncooperative game among decentralized scheduler agents as they compete with one another in the shared resources.

10 Conclusions

We have formulated and solved three optimization problems, i.e., the optimal task dispatching problems with minimized average task response time, minimized average power consumption, and minimized average cost-performance ratio, on multiple heterogeneous multiserver systems with dynamic d-speed and power management. We have also demonstrated numerical data and conducted performance comparison between dynamic management and static management of speed and power.

In this paper, each server has a known speed scheme. As a further research direction, the optimal task dispatching problem in this paper can be extended to the optimal task dispatching and speed scheme problem, in which the speed scheme of a server is also to be determined in such a way that the overall power consumption of the multiserver systems does not exceed a given power budget. This is an extremely difficult problem, since the choice of a speed scheme can be arbitrarily complicated. Even though we only consider a d-speed scheme, it still has $2d - 1$ parameters in ψ_i. By including the task arrival rate λ_i, each multiserver system has $2d$ parameters to determine, and our optimization problem has $2nd$ parameters to determine. When $d = 2$, we still have $4n$ parameters. It is conceivable that the optimization problem is very sophisticated. However, we would like to mention that when $d = 1$, i.e., for single-speed multiserver systems, the optimal load distribution and power allocation (i.e., speed determination) problem has been solved [8].

References

[1] https://en.wikipedia.org/wiki/Cloud_load_balancing
[2] http://en.wikipedia.org/wiki/CMOS
[3] http://searchcloudcomputing.techtarget.com/definition/cloud-load-balancing
[4] N. M. Al Sallami, "Load balancing in green cloud computation," *Proceedings of the World Congress on Engineering*, vol. II, 2013.
[5] Anjali, J. Grover, M. Singh, C. Singh, and H. Sethi, "A new approach for dynamic load balancing in cloud computing," *National Conference on Advances in Engineering, Technology & Management*, pp. 30-36, 2015.
[6] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, pp. 755-768, 2012.
[7] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in Computers*, vol. 82, pp. 47-111, 2011.
[8] J. Cao, K. Li, and I. Stojmenovic, "Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers," *IEEE Transactions on Computers*, vol. 63, no. 1, pp. 45-58, 2014.
[9] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE Journal on Solid-State Circuits*, vol. 27, no. 4, pp. 473-484, 1992.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2706425, IEEE Transactions on Sustainable Computing

[10] K. Dasgupta, B. Mandal, P. Dutta, J. K. Mondal, and S. Dam, "A genetic algorithm (GA) based load balancing strategy for cloud computing," *Procedia Technology*, vol. 10, pp. 340-347, 2013.

[11] B. L. D. Dhinesh and P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Applied Soft Computing*, vol. 13, pp. 2292-2303, 2013.

[12] J. Gasior and F. Seredynski, "Load balancing in cloud computing systems through formation of coalitions in a spatially generalized prisoner's dilemma game," *Third International Conference on Cloud Computing, GRIDs, and Virtualization*, pp. 201-205, 2012.

[13] S. M. Ghafari, M. Fazeli, A. Patooghy, and L. Rikhtechi, "Bee-MMT: A load balancing method for power consumption management in cloud computing," *Sixth International Conference on Contemporary Computing*, pp. 76-80, 2013.

[14] P. P. G. Gopinath and S. K. Vasudevan, "An in-depth analysis and study of load balancing techniques in the cloud computing environment," *Procedia Computer Science*, vol. 50, pp. 427-432, 2015.

[15] J. Grover and S. Katiyar, "Agent based dynamic load balancing in cloud computing," *International Conference on Human Computer Interactions*, pp. 1-6, 2013.

[16] Himanshi and S. Ahuja, "Cloud load balancing services survey and research challenges," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 6, pp. 434-441, 2015.

[17] J. Huang, R. Li, J. An, D. Ntalasha, F. Yang, and K. Li, "Energy-efficient resource utilization for heterogeneous embedded computing systems," *IEEE Transactions on Computers*, in press, 2017.

[18] Intel, *Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor – White Paper*, March 2004.

[19] N. J. Kansal and I. Chana, "Cloud load balancing techniques: A step towards green computing," *International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 238-246, 2012.

[20] S. Kapoor, "A survey on dynamic load balancing algorithms in cloud computing," *Advances in Computer Science and Information Technology*, vol. 2, no. 7, pp. 87-91, 2015.

[21] M. Katyal and A. Mishra, "A comparative study of load balancing algorithms in cloud computing environment," *International Journal of Distributed and Cloud Computing*, vol. 1, no. 2, pp. 5-14, 2013.

[22] R. Kaur and P. Luthra, "Load balancing in cloud computing," *Int'l. Conf. on Recent Trends in Information, Telecommunication and Computing*, Association of Computer Electronics and Electrical Engineers, 2014.

[23] A. Khiyaita, H. E. Bakkali, M. Zbakh, and D. E. Kettani, "Load balancing cloud computing: State of art," *National Days of Network Security and Systems*, pp. 106-109, 2012.

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

[25] F. Kong and X. Liu, "A survey on green-energy-aware power management for datacenters," *ACM Computing Surveys*, vol. 47, no. 2, article 30, 2014.

[26] K. Li, "Optimal power allocation among multiple heterogeneous servers in a data center," *Sustainable Computing: Informatics and Systems*, vol. 2, no. 1 pp. 13-22, 2012.

[27] K. Li, "Optimal load distribution for multiple heterogeneous blade servers in a cloud computing environment," *Journal of Grid Computing*, vol. 11, no. 1, pp. 27-46, 2013.

[28] K. Li, "Improving multicore server performance and reducing energy consumption by workload dependent dynamic power management," *IEEE Transactions on Cloud Computing*, vol. 4, no. 2, pp. 122-137, 2016.

[29] S. U. R. Malik, K. Bilal, S. U. Khan, B. Veeravalli, K. Li, and A. Y. Zomaya, "Modeling and analysis of the thermal properties exhibited by cyber physical data centers," *IEEE Systems Journal*, vol. 11, no. 1, pp. 163-172, 2017.

[30] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi, and J. Al-Jaroodi, "A survey of load balancing in cloud computing: challenges and algorithms," *International Symposium on Network Cloud Computing and Applications*, pp. 137-142, 2012.

[31] A.-C. Orgerie, M. D. de Assuncao, and L. Lefevre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Computing Surveys*, vol. 46, no. 4, article 47, 2014.

[32] D. Paul, W.-D. Zhong, and S. K. Bose, "Energy efficiency aware load distribution and electricity cost volatility control for cloud service providers," *Journal of Network and Computer Applications*, vol. 59, pp. 185-197, 2016.

[33] M. Rahman, S. Iqbal, and J. Gao, "Load balancer as a service in cloud computing," *IEEE 8th International Symposium on Service Oriented System Engineering*, pp. 204-211, 2014.

[34] Y. Sahu, R. K. Pateriya, and R. K. Gupta, "Cloud server optimization with load balancing and green computing techniques using dynamic compare and balance algorithm," *5th International Conference on Computational Intelligence and Communication Networks*, pp. 527-531, 2013.

[35] P. M. Shameem and R. S. Shaji, "A methodological survey on load balancing techniques in cloud computing," *International Journal of Engineering and Technology*, vol. 5, no. 5, pp. 3801-3812, 2013.

[36] B. A. Shirazi, A. R. Hurson, and K. M. Kavi, eds., *Scheduling and Load Balancing in Parallel and Distributed Systems*, IEEE Computer Society Press, Los Alamitos, California, 1995.

[37] A. Singh, K. Dutta, and H. Gupta, "A survey on load balancing algorithms for cloud computing," *International Journal of Computer Application*, vol. 6, no. 4, pp. 66-72, 2014.

[38] A. Singh, D. Juneja, and M. Malhotra, "Autonomous agent based load balancing algorithm in cloud computing," *Procedia Computer Science*, vol. 45, pp. 832-841, 2015.

[39] R. K. Srinivasan, V. Suma, and V. Nedu, "An enhanced load balancing technique for efficient load distribution in cloud-based IT industries," *International Symposium on Intelligent Informatics*, pp. 479-485, 2012.

[40] Y. Tian, C. Lin, and K. Li, "Managing performance and power consumption tradeoff for multiple heterogeneous servers in cloud computing," *Cluster Computing*, vol. 17, no. 3, pp. 943-955, 2014.

[41] Z. Tong, Z. Xiao, K. Li, and K. Li, "Proactive scheduling in distributed computing – A reinforcement learning approach," *Journal of Parallel and Distributed Computing*, vol. 74, no. 7, pp. 2662-2672, 2014.

[42] Z. Xiao, P. Liang, Z. Tong, K. Li, S. U. Khan, and K. Li, "Self-adaptation and mutual adaptation for distributed scheduling in benevolent clouds," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 5, 12 pp., 2017.

[43] Z. Xiao, Z. Tong, K. Li, and K. Li, "Learning non-cooperative game for load balancing under self-interested distributed environment," *Applied Soft Computing*, vol. 52, pp. 376-386, 2017.

[44] B. Yang, Z. Li, S. Chen, T. Wang, and K. Li, "A stackelberg game approach for energy-aware resource allocation in data centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 12, pp. 3646-3658, 2016.

[45] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," *Proceedings of the 41st Design Automation Conference*, pp. 868-873, 2004.

Keqin Li is a SUNY Distinguished Professor of computer science. He is also a Distinguished Professor of Chinese National Recruitment Program of Global Experts (1000 Plan) at Hunan University, China. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU-GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyber-physical systems. He has published over 490 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, and *IEEE Transactions on Sustainable Computing*. He is an IEEE Fellow.