NAIST-IS-1451052

Master's Thesis

Performance Analysis of Power Management Scheme for Data Centers

Tomoyuki Sakata

March 14, 2016

Graduate School of Information Science
Nara Institute of Science and Technology

A Master's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
MASTER of ENGINEERING

Tomoyuki Sakata

Thesis Committee:

Professor Shoji Kasahara	(Supervisor)
Professor Kazushi Ikeda	(Co-supervisor)
Associate Professor Masahiro Sasabe	(Co-supervisor)

Performance Analysis of Power Management Scheme for Data Centers*

Tomoyuki Sakata

Abstract

One of approaches to reducing energy consumption in data centers is to power down a group of servers according to the utilization rate, or the number of jobs in the system. In this thesis, we consider a power management scheme for distributed parallel processing over clusters of servers. For this scheme, a part of servers in each cluster are turned on/off according to the state of a background process, or the number of jobs in the system. In the former case, we model the system as a multi-server queue in which the service time of a job depends on the state of a background process at the beginning of the job service. In the latter case, on the other hand, we consider a multi-server queueing system where the service time of a job depends on the number of jobs in the system. For both systems, we analyze the distribution of the number of jobs in the system, deriving the mean job-response time and mean amount of energy consumption. In the latter case, we further consider the product of those two measures. In numerical examples, we investigate how performance measures are affected by the background process or thresholds which manage energy saving level.

Keywords:

Data center, power management, multi-server queue, performance evaluation, matrix analytic method.

*Master's Thesis, Graduate School of Information Science,
Nara Institute of Science and Technology, NAIST-IS-1451052, March 14, 2016.

Contents

List of Figures

1. Introduction

1.1. Background

Recently, cloud computing has attracted considerable attention, and various kinds of computing services such as virtual machines and MapReduce software framework are provided by data centers [1]. A data center contains a large number of server machines, resulting in high energy consumption [8]. With the increase in cloud computing demand, the number of data centers is growing rapidly, and the amount of energy consumption for data centers is extremely huge [2]. Therefore, considerable research efforts have been devoted to developing schemes which can save energy without degrading job-processing performance.

1.2. Related Work

There exist much literature which concerns strategies of efficient energy saving for data centers. A typical scheme for reducing energy consumption in data centers is to manage the power of server machines according to demand for computing resources. In [10], the authors propose a scheme of server-power management for a data center, with which the number of running servers is varied according to the number of jobs in the system. If the number of waiting jobs exceeds a predefined threshold, all the servers are turned on. If the number of servers busy for job processing is below another threshold, a given number of servers are turned off. In [10], the trade-off between job-waiting time and energy consumption is analyzed by an M/M/c queue with threshold-based on/off control.

The authors of [4] consider the optimal policy for power management of data centers. They model a data center as an M/M/c-type queue, where server states consist of busy, idle, and several sleep states. Taking into account the transition

time from a sleep state to busy one, the authors investigate by simulation the optimal policy such that the product of the mean job-response time and power consumption is minimized.

In [3], the authors propose Berkeley Energy Efficient MapReduce (BEEMR), an efficient power management scheduling for MapReduce-type job processing. In BEEMR, servers in a data center are divided into two grouping zones, an interactive zone and a batch zone. The servers in the interactive zone are always turned on and serve small-sized jobs. On the other hand, the servers in the batch zone process huge-sized jobs that are insensitive to the response time. BEEMR controls the power of servers in the batch zone so that the amount of power consumption is reduced. The authors in [3] investigate the performance of BEEMR by simulation and on-site practical experiments. In [5], the performance of BEEMR is investigated by queueing theoretical approach.

1.3. Research Objective and Outline of Thesis

In this thesis, we consider a power-management scheme for data centers with server clusters. We focus on a data center accommodating a large number of server clusters, each of which consists of several server machines, providing parallel distributed computing service. The data center alternates two power-operation modes: normal-operation and power-saving. The power of server machines is managed in a cluster-based manner. When the data center is in normal-operation mode, all the servers of all clusters are powered on. When the data center switches to power-saving mode, a part of servers in each cluster are powered off after completing the existing job. In both operation modes, a job is served by a cluster according to processor sharing discipline. Therefore, the service rate of a cluster in power-saving mode is smaller than that in normal mode.

In order to investigate the performance of the cluster-based power management scheme, for data centers, we consider two multi-server queueing systems. In the first system, the power-operation modes alternate according to the state of a background process. In the second system, on the other hand, we consider the power management which the power-operation mode depends on the number of jobs in the system. In both the systems, a server of the queueing model

corresponds to a cluster of machines in the data center. For each queueing system, we construct a trivariate continuous-time Markov chain for the system, deriving the steady-state probability vector by matrix geometric method. We consider the mean job-response time and mean amount of energy consumption. For the second system, we further consider the product of them as one of performance measures. In both systems, we investigate how the performance measures are affected by design parameters which manage energy saving level.

This thesis is organized as follows. In Chapter 2, we consider a multi-server queue in which the service time of a job depends on the state of a background process at the beginning of the job service. We analyze the joint distribution of the number of jobs in the system and the state of the background process, deriving the mean job-response time and mean amount of energy consumption. In numerical examples, we investigate how the mean job-response time and energy consumption are affected by energy saving level and the number of clusters.

In Chapter 3, we consider the power-management scheme with which a part of servers in each cluster are turned on/off according to the number of jobs in the system. We model the system as a multi-server queue in which the service time of a job depends on the number of jobs in the system at the beginning of the job service. We analyze the distribution of the number of jobs in the system, deriving the mean job-response time, mean amount of energy consumption, and the product of those two measures. In numerical examples, we investigate how the mean job-response time and energy consumption are affected by thresholds which manage energy saving level.

Chapter 4 gives conclusion of the thesis.

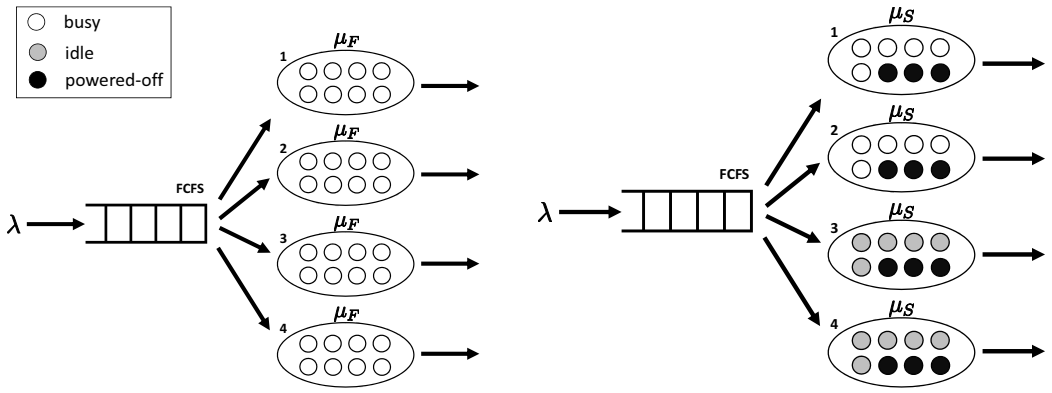
2. Multi-Server Queue with Job Service Time Depending on a Background Process

2.1. Queueing Model

The system consists of c homogeneous server clusters and a single queue with infinite capacity. (See Figure 2.1.) Each cluster accommodates several servers, and the power of some servers in a cluster is controlled according to the power-management policy described later in this section. In Figure 2.1, the number of clusters is four, each cluster has eight servers, and the number of power-controlled servers in each cluster is three.

Jobs arrive at the system according to a Poisson process with rate λ. Each cluster serves a job in a parallel-distributed processing manner. That is, only one job can enter an idle cluster. The service rate of the job depends on the number of powered-on servers in the cluster. We consider two power-control modes for clusters, Slow (S) and Fast (F). The mode S describes the power-saving mode in which a part of server machines are turned off for energy saving (Figure 2.1b), and hence the resulting service rate of a cluster is low. When the mode is F, on the other hand, all the server machines composing a cluster are turned on (Figure 2.1a) and the resulting service rate of the server is greater than that in mode S. We call F normal-operation mode hereafter. We assume that the service time of a job in mode S (resp. F) follows an exponential distribution with rate μ_S (resp. μ_F). Hereafter, a job served with rate $\mu_S > 0$ and that with rate $\mu_F > 0$ are called S job and F job, respectively.

The system mode changes from S to F and vice versa according to the state of



(a) Normal-operation mode. (b) Power-saving mode.

Figure 2.1.: System model.

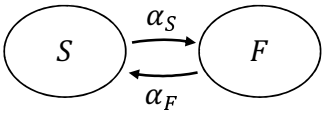


Figure 2.2.: Background process.

a background process. (See Figure 2.2.) We assume that the service rate of a job is governed by the system mode at its service initiation point. The background process is continuous-time Markov chain with two states, S and F, independent of the arrival process. The state-transition rate from S to F and that from F to S is given by $\alpha_S > 0$ and $\alpha_F > 0$, respectively.

When a job enters a server for its cluster, its service time depends on the state of the background process. If the background process is in state S (resp. F) at the service initiation point, the service time of the job follows an exponential distribution with rate μ_S (resp. μ_F). In the following, $\mu_S \leq \mu_F$. We also assume that when the background process switches from S to F (and vice versa), the service rate of the existing job remains the same as that at its service initiation point.

2.2. Analysis

We define $N(t)$ as the number of jobs in the system at time t. Let $S(t)$ and $F(t)$ denote the numbers of S jobs and F jobs at t, respectively. We denote $J(t)\,(\in \{S, F\})$ as the mode of power-management at t. $F(t)$ can be expressed with $N(t)$ and $S(t)$ by

$$F(t) = \min(N(t), c) - S(t),$$

From the assumptions, $\{(N(t), S(t), J(t)) : t \geq 0\}$ is a trivariate continuous-time Markov chain with state space $\mathbb{F}$, where $\mathbb{F}$ is given by

$$\mathbb{F} = \mathbb{N} \cup \{0\} \times \{0, 1, \ldots, c\} \times \{S, F\}.$$

Let $\boldsymbol{Q}$ denote the infinitesimal generator of the Markov chain $\{N(t), S(t), J(t) : t \geq 0\}$, whose states are arranged in lexicographic order. Then, $\boldsymbol{Q}$ is given by

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{B}_0 & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots \\ \boldsymbol{B}_1 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots \\ \boldsymbol{O} & \boldsymbol{A}_2 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & \boldsymbol{O} & \boldsymbol{O} & \cdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_2 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & \boldsymbol{O} & \cdots \\ & & & \ddots & \ddots & \ddots & \end{bmatrix}. \tag{2.1}$$

In what follows, we describe the details of block matrices $\boldsymbol{B}$, $\boldsymbol{B}_0$, $\boldsymbol{B}_1$, $\boldsymbol{A}_0$, $\boldsymbol{A}_1$, and $\boldsymbol{A}_2$ in (2.1). Hereafter, i is an integer such that the inequality $k(k+1) < i \leq (k+1)(k+2)$ holds for any $k \in [0, c]$, and $[x]$ is the largest integer not greater than x.

(a) $c(c+1) \times c(c+1)$ matrix $\boldsymbol{B}$

(i) For odd i,

$$[\boldsymbol{B}]_{ij} = \begin{cases} \alpha_S, & j = i+1, \\ \lambda, & j = i+2k+4, \\ d(k+1, i+1)\,\mu_F, & j = i-2k, \\ -d(k, i-1)\,\mu_S, & j = i-2k-2, \\ -\alpha_S - \lambda - d(k+1, i+1)\,\mu_F + d(k, i-1)\,\mu_S, & j = i, \\ 0, & \text{otherwise}, \end{cases}$$

where $d(n,m) = \dfrac{n(n+1)-m}{2}$.

(ii) For even i,

$$[\boldsymbol{B}]_{ij} = \begin{cases} \alpha_F, & j = i-1, \\ \lambda, & j = i+2k+2, \\ d(k+1,i)\,\mu_F, & j = i-2k, \\ -d(k,i-2)\mu_S, & j = i-2k-2, \\ -\alpha_F - \lambda - d(k+1,i)\,\mu_F + d(k,i-2)\mu_S, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

(b) $c(c+1) \times 2(c+1)$ matrix $\boldsymbol{B}_0$

(i) For odd i,

$$[\boldsymbol{B}_0]_{ij} = \begin{cases} \lambda, & j = i - c(c-1) + 2 \text{ and } j \neq 1, \\ 0, & \text{otherwise.} \end{cases}$$

(ii) For even i,

$$[\boldsymbol{B}_0]_{ij} = \begin{cases} \lambda, & j = i - c(c-1), \\ 0, & \text{otherwise.} \end{cases}$$

(c) $2(c+1) \times c(c+1)$ matrix $\boldsymbol{B}_1$

$$[\boldsymbol{B}_1]_{ij} = \begin{cases} \left(c - \left[\dfrac{i-1}{2}\right]\right)\mu_F, & j = i + c(c-1), \\ \left[\dfrac{i-1}{2}\right]\mu_S, & j = i + c(c-1) - 2, \\ 0, & \text{otherwise.} \end{cases}$$

(d) $2(c+1) \times 2(c+1)$ matrix $\boldsymbol{A}_0$

$$[\boldsymbol{A}_0]_{ij} = \begin{cases} \lambda, & j = i, \\ 0, & j \neq i. \end{cases}$$

(e) $2(c+1) \times 2(c+1)$ matrix $\boldsymbol{A}_1$

(i) For odd i,

$$[\boldsymbol{A}_1]_{ij} = \begin{cases} \alpha_S, & j = i+1, \\ -\alpha_S - \lambda - \left[\dfrac{i-1}{2}\right]\mu_S - \left(c - \left[\dfrac{i-1}{2}\right]\right)\mu_F, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

(ii) For even i,

$$[\boldsymbol{A}_1]_{ij} = \begin{cases} \alpha_F, & j = i-1, \\ -\alpha_F - \lambda - \left[\dfrac{i-1}{2}\right]\mu_S - \left(c - \left[\dfrac{i-1}{2}\right]\right)\mu_F, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

(f) $2(c+1) \times 2(c+1)$ matrix $\boldsymbol{A}_2$

(i) For odd i,

$$[\boldsymbol{A}_2]_{ij} = \begin{cases} \dfrac{i-1}{2}\mu_S, & j = i, \\ \left(c - \dfrac{i-1}{2}\right)\mu_F, & j = i+2, \\ 0, & \text{otherwise.} \end{cases}$$

(ii) For even i,

$$[\boldsymbol{A}_2]_{ij} = \begin{cases} \left(c - \dfrac{i-2}{2}\right)\mu_F, & j = i, \\ \dfrac{i-2}{2}\mu_S, & j = i-2, \\ 0, & \text{otherwise.} \end{cases}$$

We define the steady-state probability as

$$\pi(i,j,k) = \lim_{t\to\infty} \Pr\{N(t) = i, S(t) = j, J(t) = k\}, \quad (i,j,k) \in \mathbb{F}.$$

We also define the following notations.

$$\begin{aligned} \boldsymbol{\pi}_{-1} = (&\pi(0,0,S), \pi(0,0,F), \pi(1,0,S), \pi(1,0,F), \pi(1,1,S), \pi(1,1,F), \ldots, \\ &\pi(c-1,0,S), \pi(c-1,0,F), \pi(c-1,1,S), \pi(c-1,1,F), \ldots, \\ &\pi(c-1,c-1,S), \pi(c-1,c-1,F)), \\ \boldsymbol{\pi}_i = (&\pi(c+i,0,S), \pi(c+i,0,F), \pi(c+i,1,S), \pi(c+i,1,F), \ldots, \\ &\pi(c+i,c,S), \pi(c+i,c,F)), \quad i \geq 0. \end{aligned}$$

Let

$$\boldsymbol{\pi} = (\boldsymbol{\pi}_{-1}, \boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \cdots).$$

$\boldsymbol{\pi}$ is the steady-state probability vector which satisfies

$$\begin{cases} \boldsymbol{\pi}\boldsymbol{Q} = \boldsymbol{0}, \\ \boldsymbol{\pi}\boldsymbol{e} = 1. \end{cases} \tag{2.2}$$

From (2.1), this continuous-time Markov chain is a quasi birth-and-death process. The steady-state probability vector $\boldsymbol{\pi}$ can be calculated by matrix-analytic method [6].

When $i \geq 1$, from (2.1), (2.2), $\boldsymbol{\pi}_i$ satisfies the following equation.

$$\boldsymbol{\pi}_{i-1}\boldsymbol{A}_0 + \boldsymbol{\pi}_i\boldsymbol{A}_1 + \boldsymbol{\pi}_{i+1}\boldsymbol{A}_2 = \boldsymbol{0}. \tag{2.3}$$

There is a rate matrix $\boldsymbol{R}$ such that $\boldsymbol{\pi}_i = \boldsymbol{\pi}_{i-1}\boldsymbol{R}$, and they have a matrix geometric form,

$$\boldsymbol{\pi}_i = \boldsymbol{\pi}_0\boldsymbol{R}^i. \tag{2.4}$$

Substituting (2.4) into (2.3),

$$\boldsymbol{\pi}_0\boldsymbol{R}^{i-1}\boldsymbol{A}_0 + \boldsymbol{\pi}_0\boldsymbol{R}^i\boldsymbol{A}_1 + \boldsymbol{\pi}_0\boldsymbol{R}^{i+1}\boldsymbol{A}_2 = \boldsymbol{0}.$$

Therefore,

$$\boldsymbol{A}_0 + \boldsymbol{R}\boldsymbol{A}_1 + \boldsymbol{R}^2\boldsymbol{A}_2 = \boldsymbol{O}. \tag{2.5}$$

In order to calculate $\boldsymbol{R}$ in (2.5), considering

$$\begin{cases} \boldsymbol{R}^{(0)} & = \boldsymbol{O}, \\ \boldsymbol{R}^{(n+1)} & = -\boldsymbol{A}_2\boldsymbol{A}_1^{-1} - (\boldsymbol{R}^{(n)})^2\boldsymbol{A}_0\boldsymbol{A}_1^{-1}, \end{cases}$$

we should compute $\boldsymbol{R} = \lim_{n\to\infty} \boldsymbol{R}^{(n)}$. In practical calculation, we choose the iteration count $\hat{N}$ satisfying the following inequality.

$$\hat{N} = \min\{n \geq 1 : \|\boldsymbol{R}^{(n)} - \boldsymbol{R}^{(n-1)}\|_{\max} < \varepsilon\},$$

where $\|\cdot\|_{\max}$ is the max norm such that $\|\boldsymbol{K}\|_{\max} = \max_{i,j} |[\boldsymbol{K}]_{ij}|$ for matrix $\boldsymbol{K}$.

$\boldsymbol{\pi}_{-1}$,$\boldsymbol{\pi}_0$,$\boldsymbol{\pi}_1$ satisfy the following equations.

$$\begin{aligned} \boldsymbol{\pi}_{-1}\boldsymbol{B} + \boldsymbol{\pi}_0\boldsymbol{B}_1 &= \boldsymbol{0}, \\ \boldsymbol{\pi}_{-1}\boldsymbol{B}_0 + \boldsymbol{\pi}_0\boldsymbol{A}_1 + \boldsymbol{\pi}_1\boldsymbol{A}_2 &= \boldsymbol{0}. \end{aligned}$$

The normalization condition is also given by

$$\boldsymbol{\pi}_{-1}\boldsymbol{e}_0 + \sum_{j=0}^{\infty} \boldsymbol{\pi}_j \boldsymbol{e}_1 = 1,$$

where $\boldsymbol{e}_0$ and $\boldsymbol{e}_1$ are suitably dimensioned column vectors of 1s. Noting that

$$\sum_{j=0}^{\infty} \boldsymbol{\pi}_j \boldsymbol{e}_1 = \boldsymbol{\pi}_0 \sum_{j=1}^{\infty} \boldsymbol{R}^{j-1}\boldsymbol{e}_1 = \boldsymbol{\pi}_0(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{e}_1,$$

and $\boldsymbol{\pi}_1 = \boldsymbol{\pi}_0\boldsymbol{R}$, we can write

$$(\boldsymbol{\pi}_{-1}, \boldsymbol{\pi}_0) \begin{bmatrix} \boldsymbol{e}_0 & \boldsymbol{B}^* & \boldsymbol{B}_0 \\ (\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{e}_1 & \boldsymbol{B}_1^* & \boldsymbol{A}_1 + \boldsymbol{R}\boldsymbol{A}_2 \end{bmatrix} = [1, \boldsymbol{0}],$$

where $\boldsymbol{B}^*$ and $\boldsymbol{B}_1^*$ are matrices $\boldsymbol{B}$ and $\boldsymbol{B}_1$ with its first column eliminated, and $\boldsymbol{0}$ is $(c+1)(c+2)-1$ dimensioned column vector of 0s.

Therefore, if we solve $\boldsymbol{\pi}_{-1}$, $\boldsymbol{\pi}_0$, $\boldsymbol{R}$, we can calculate stationary state probability of all states.

In terms of the system stability, we have the following theorem.

Theorem 2.2.1. *([7], p. 411, (9.36)) We assume that* $2(c+1)$ *dimensional square matrix* $\boldsymbol{A} = \boldsymbol{A}_0 + \boldsymbol{A}_1 + \boldsymbol{A}_2$ *satisfies* $\boldsymbol{\pi}_A\boldsymbol{A} = \boldsymbol{0}$ *and* $\boldsymbol{\pi}_A\boldsymbol{e}_1 = 1$.

Then, the stability condition for the system is

$$\boldsymbol{\pi}_A\boldsymbol{A}_0\boldsymbol{e}_1 < \boldsymbol{\pi}_A\boldsymbol{A}_2\boldsymbol{e}_1.$$

In our case, we can conjecture the following stability condition (see Appendix A).

$$\lambda < \frac{c\mu_S\mu_F\left(\alpha_S^2 + 2\alpha_S\alpha_F + \alpha_F^2 + \alpha_S\mu_F + \alpha_F\mu_S\right)}{(\alpha_S+\alpha_F)(\alpha_S\mu_S + \alpha_F\mu_F + \mu_S\mu_F)}. \tag{2.6}$$

We consider two performance measures: the mean job-response time and mean amount of energy consumption. The mean number of jobs in the system and that in queue is given by

$$\mathrm{E}[L] = \sum_{(i,j,k)\in\mathbb{F}} i\pi(i,j,k), \tag{2.7}$$

Let N^* denote the number to truncate the calculation of infinite summation in right side of (3.3). N^* is defined by

$$N^* = \min\{n \geq 0 : |[n\boldsymbol{\pi}_n - (n-1)\boldsymbol{\pi}_{n-1}]\boldsymbol{e}_1| < \varepsilon\}$$

Using Little's law, the mean job-response time in the system $\mathrm{E}[T]$ is given by

$$\mathrm{E}[T] = \mathrm{E}[L]/\lambda.$$

Let E denote the mean amount of energy consumption per unit time. E can be expressed as

$$E = \sum_{(i,j,k)\in\mathbb{F}} \pi(i,j,k)\{j\mu_S + \min(c-j, i-j)\mu_F + \max(0, c-i)\kappa\mu_k\},$$

where κ is the ratio of the amount of energy consumption of a single idle server to that of a single busy server.

2.3. Numerical Examples

In this section, we show some numerical examples.

Let ζ denote the power-saving level defined by $\zeta = \mu_S/\mu_F$, the ratio of the service rate of power-saving mode to that of normal-operation mode. A small ζ indicates that the number of worker machines turned off in power-saving mode is large. In other words, a small ζ implies a small amount of energy consumption for the system. Note that when the system is in power-saving mode, all the servers keep running for $\zeta = 1$, whereas all the servers are turned off for $\zeta = 0$*.

In the following, we set $\kappa = 170/240$ according to [10]. We consider the case of $c = 20$ and $\mu_F = 5$ throughout the section. We only consider the case of $\alpha_S = \alpha_F = 1$.

*The case of $\zeta = 0$ is equivalent to the model in [10].

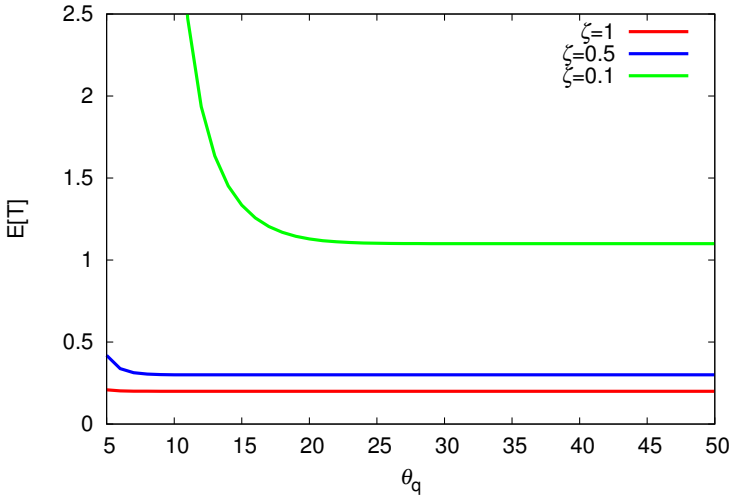


Figure 2.3.: The mean job-response time E[T] vs. c.

Figure 2.3 illustrates the mean job-response time E[T]. The horizontal axis is the number of clusters c, and E[T]'s for $\zeta = 0.1$, 0.5, 1 are plotted. Note that $\zeta = 1$ under $\alpha_S = \alpha_F$ corresponds to the case of an M/M/c with service rate μ_F. We observe from the figure that when c increases, E[T]'s for the three cases decrease and converge to some constants. We also observe that E[T] increases with the decrease in ζ. A remarkable point here is that the discrepancy between $\zeta = 1$ and $\zeta = 0.5$ is significantly smaller than that between $\zeta = 1$ and $\zeta = 0.1$. When $\zeta = 0.5$, a half of servers in a cluster are powered off in power-saving mode. This figure shows that energy-saving level of $\zeta = 0.5$ does not degrade the job-response time.

Figure 2.4 represents E against the number of clusters c. In this figure, the amount of energy consumption increases linearly for any ζ, as expected. We also observe that E for $\zeta = 1$ is the largest for any c, and that E becomes small with the decrease in ζ. Note that the power-saving level of $\zeta = 0.5$ effectively reduces E even for a small c. This result suggests that the power-saving level of $\zeta = 0.5$ is effective both for reducing the energy consumption and for keeping the job-response time small.

We also investigated the job-response time and energy consumption in cases of different λ's, observing the same tendency as Figures 2.3 and 2.4. This suggests

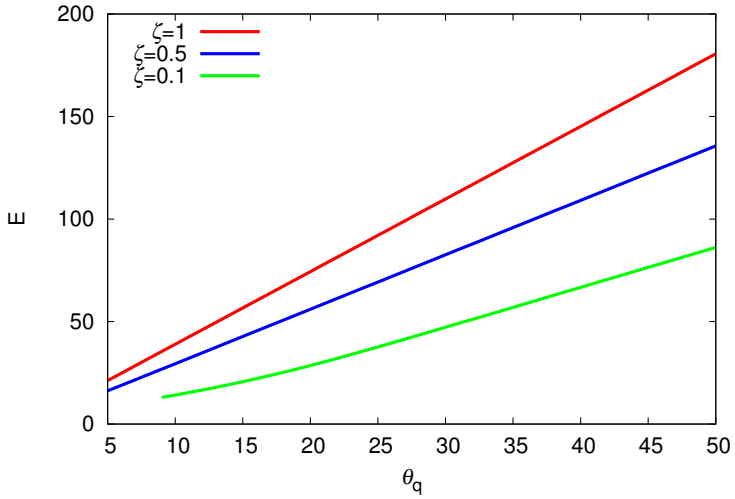


Figure 2.4.: The mean energy consumption E vs. c.

that turning a half of servers in a cluster off is effective for keeping the mean job-response time small.

2.4. Conclusion

In this chapter, we considered a queueing model for data centers with BEEMR-like energy-saving management mechanism. We modeled a data center as a multiple-server queue with job service depending on a background process. Using matrix-analytic method, we derived the joint distribution of the number of jobs in the system and the state of the background process, yielding the mean job-response time and mean amount of energy consumption as performance measures. Numerical examples showed that the amount of energy consumption grows linearly with the increase in the number of clusters. We also confirmed that turning a half of servers in a cluster off is effective for keeping the mean job-response time small.

3. Queueing System with Power Management Scheme Depending on the Number of Jobs in the System

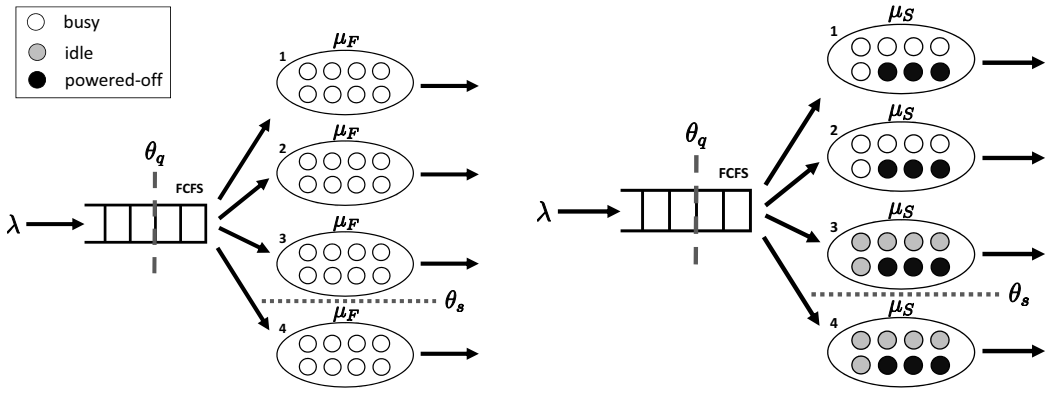
3.1. Queueing Model

In this chapter, the system mode changes from S to F and vice versa according to the transition of the number of jobs in the system. We consider the following power management scheme using two thresholds θ_s, θ_q ($0 < \theta_s \leq c < \theta_q$), which is originally considered in [10]. Suppose that the number of jobs in the system is zero at system initiation and that the system is in power-saving mode S.

1. Until the number of jobs in the system become more than θ_q, the system stays in mode S and the service rate of each cluster is μ_S.

2. When a new job arrives at the system and the number of jobs in the system reaches θ_q, the system mode changes from S to normal-operation mode F and the next served job is processed with service rate μ_F. Note that the service rate of the existing jobs remains μ_S.

3. When the number of jobs in the system becomes lower than θ_s, the system mode changes from F to power-saving mode S, and newly arriving jobs are served with rate μ_S. Then, steps 1 to 3 are repeated.



(a) Normal-operation mode. (b) Power-saving mode.

Figure 3.1.: System model.

3.2. Analysis

As it is for the queueing model explained in Chapter 2, $\{(N(t), S(t), J(t)) : t \geq 0\}$ is a trivariate continuous-time Markov chain with state space $\mathbb{F}$, where $\mathbb{F}$ is given by

$$\mathbb{F} = \mathbb{N} \cup \{0\} \times \{0, 1, \ldots, c\} \times \{S, F\}.$$

Let $\boldsymbol{Q}$ denote the infinitesimal generator of the Markov chain $\{N(t), S(t), J(t) : t \geq 0\}$, whose states are arranged in lexicographic order. Then, $\boldsymbol{Q}$ is given by

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{B}_0 & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots \\ \boldsymbol{B}_1 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots \\ \boldsymbol{O} & \boldsymbol{A}_2 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & \boldsymbol{O} & \boldsymbol{O} & \cdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_2 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & \boldsymbol{O} & \cdots \\ & & & \ddots & \ddots & \ddots & \end{bmatrix}. \tag{3.1}$$

In what follows, we describe the details of block matrices $\boldsymbol{B}$, $\boldsymbol{B}_0$, $\boldsymbol{B}_1$, $\boldsymbol{A}_0$, $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ in (3.1). Hereafter, we define l as

$$l = (c+1)(c+2) + 2(c+1)(\theta_q - c - 1),$$

and k as the number of jobs corresponding to an element $[\cdot]_{ij}$ in each matrix, that is to say, k is the integer such that the inequality $k(k+1) < i \leq (k+1)(k+2)$

holds for any $i \in \{1, 2, \ldots, (c+1)(c+2)\}$, and

$$k = c + 1 + \left[\frac{i - (c+1)(c+2)}{2(c+1)} \right],$$

for $(c+1)(c+2) < i \leq l$. We also define n as the number of the S jobs.

(a) $l \times l$ matrix $\boldsymbol{B}$

(i) The odd i case

- If $j = i + 2k + 4, k < c$, or $j = i + 2c + 2, c \leq k < \theta_q$, then

$$[\boldsymbol{B}]_{ij} = \lambda.$$

- If $j = i - 2k, k < c$, then

$$[\boldsymbol{B}]_{ij} = \frac{(k+1)(k+2) - 1 - i}{2} \mu_F.$$

- If $j = i - 2k - 2, k < c$, then

$$[\boldsymbol{B}]_{ij} = \frac{i - k(k+1) - 1}{2} \mu_S.$$

- If $j = i - 2c - 2, k = c$, or $j = i - 2c - 4, c < k < \theta_q$, then

$$[\boldsymbol{B}]_{ij} = n\mu_S.$$

- If $j = i - 2c, k = c$, or $j = i - 2c, c < k < \theta_q$, then

$$[\boldsymbol{B}]_{ij} = (c - n)\mu_F.$$

- If $j = i, k < c$, then

$$[\boldsymbol{B}]_{ij} = -\lambda - \frac{(k+1)(k+2) - 1 - i}{2} \mu_F - \frac{i - k(k+1) - 1}{2} \mu_S.$$

- If $j = i, c \leq k < \theta_q$, then

$$[\boldsymbol{B}]_{ij} = -\lambda - (c - n)\mu_F - n\mu_S.$$

- For others, $[\boldsymbol{B}]_{ij} = 0$.

(ii) The even i case

- If $j = i + 2k + 4,\ k < \theta_s - 1$, or $j = i + 2k + 3,\ k = \theta_s - 1$, or $j = i + 2k + 2,\ \theta_s \leq k < c$, or $j = i + 2c + 2,\ c \leq k < \theta_q$, then

$$[\boldsymbol{B}]_{ij} = \lambda.$$

- If $j = i - 2k, k < c$, then

$$[\boldsymbol{B}]_{ij} = \frac{(k+1)(k+2) - i}{2}\mu_F.$$

- If $j = i - 2k - 2, k < c$, then

$$[\boldsymbol{B}]_{ij} = \frac{i - k(k+1) - 2}{2}\mu_S.$$

- If $j = i - 2c - 2, k = c$, or $j = i - 2c - 4, c < k < \theta_q$, then

$$[\boldsymbol{B}]_{ij} = n\mu_S.$$

- If $j = i - 2c, k = c$, or $j = i - 2c - 2, c < k < \theta_q$, then

$$[\boldsymbol{B}]_{ij} = (c - n)\mu_F.$$

- If $j = i, k < c$, then

$$[\boldsymbol{B}]_{ij} = -\lambda - \frac{(k+1)(k+2) - i}{2}\mu_F - \frac{i - k(k+1) - 2}{2}\mu_S.$$

- If $j = i, c \leq k < \theta_q$, then

$$[\boldsymbol{B}]_{ij} = -\lambda - (c - n)\mu_F - n\mu_S.$$

- For others, $[\boldsymbol{B}]_{ij} = 0$.

(b) $l \times 2(c+1)$ matrix $\boldsymbol{B}_0$

$$[\boldsymbol{B}_0]_{ij} = \begin{cases} \lambda, & j = i - (l - 2(c-1)), \\ 0, & \text{otherwise.} \end{cases}$$

(c) $2(c+1) \times l$ matrix $\boldsymbol{B}_1$

(i) For odd i,

$$[\boldsymbol{B}_1]_{ij} = \begin{cases} \left(c - \left\lceil \dfrac{i-1}{2} \right\rceil\right) \mu_F, & j = i + l - 2c - 1, \\ \left\lceil \dfrac{i-1}{2} \right\rceil \mu_S, & j = i + l - 2c - 3, \\ 0, & \text{otherwise.} \end{cases}$$

(ii) For even i,

$$[\boldsymbol{B}_1]_{ij} = \begin{cases} \left(c - \left\lceil \dfrac{i-1}{2} \right\rceil\right) \mu_F, & j = i + l - 2c - 2, \\ \left\lceil \dfrac{i-1}{2} \right\rceil \mu_S, & j = i + l - 2c - 4, \\ 0, & \text{otherwise.} \end{cases}$$

(d) $2(c+1) \times 2(c+1)$ matrix $\boldsymbol{A}_0$

$$[\boldsymbol{A}_0]_{ij} = \begin{cases} \lambda, & j = i, \\ 0, & j \neq i. \end{cases}$$

(e) $2(c+1) \times 2(c+1)$ matrix $\boldsymbol{A}_1$

$$[\boldsymbol{A}_1]_{ij} = \begin{cases} -\lambda - \left\lceil \dfrac{i-1}{2} \right\rceil \mu_S - \left(c - \left\lceil \dfrac{i-1}{2} \right\rceil\right) \mu_F, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

(f) $2(c+1) \times 2(c+1)$ matrix $\boldsymbol{A}_2$

$$[\boldsymbol{A}_2]_{ij} = \begin{cases} \left(c - \left\lceil \dfrac{i-1}{2} \right\rceil\right) \mu_F, & j = i, \\ \left\lceil \dfrac{i-1}{2} \right\rceil \mu_S, & j = i - 2, \\ 0, & \text{otherwise.} \end{cases}$$

We define the steady-state probability as

$$\pi(i,j,k) = \lim_{t \to \infty} \Pr\{N(t) = i, S(t) = j, J(t) = k\}, \quad (i,j,k) \in \mathbb{F}.$$

$\boldsymbol{\pi}$ is the steady-state probability vector which satisfies $\boldsymbol{\pi Q} = \mathbf{0}$ and $\boldsymbol{\pi e} = 1$. From (3.1), this continuous-time Markov chain is a quasi birth-and-death process. The steady-state probability vector $\boldsymbol{\pi}$ can be calculated by matrix-analytic method [6].

From Theorem 2.2.1, we obtain the following stability condition.

$$\lambda < c\mu_F. \tag{3.2}$$

Finally, we consider two performance measures: the mean job-response time and mean amount of energy consumption as in Chapter 2. The mean number of jobs in the system is given by

$$\mathrm{E}[L] = \sum_{(i,j,k)\in\mathbb{F}} i\pi(i,j,k). \tag{3.3}$$

Using Little's law, the mean job-response time $\mathrm{E}[T]$ is given by $\mathrm{E}[T] = \mathrm{E}[L]/\lambda$.

Let E denote the mean amount of energy consumption per unit time. E can be expressed as

$$E = \sum_{(i,j,k)\in\mathbb{F}} \pi(i,j,k)\{j\mu_S + \min(c-j, i-j)\mu_F + \max(0, c-i)\kappa\mu_k\},$$

where κ is the ratio of the amount of energy consumption of a single idle server to that of a single busy server.

We usually need to consider both the mean job-response time and mean amount of energy consumption simultaneously. Optimizing the trade-off between these quantities, however, is a difficult problem. To capture this trade-off, we consider the Energy-Response time Product (ERP) metric [4]. The ERP is given by

$$\mathrm{ERP} = \mathrm{E}[T] \cdot E/c.$$

Minimizing ERP can be seen as improving the efficiency in terms of both the mean job-response time and mean amount of energy consumption.

3.3. Numerical Examples

3.3.1. Impact of Thresholds on Performance

Figures 3.2 and 3.3 illustrate the mean job-response time $\mathrm{E}[T]$ and the mean amount of energy consumption E, respectively. The horizontal axis is the threshold θ_q, and cases of $\theta_s = 1$, 5, 10, 15, 20 are plotted. Note that $\theta_s = 1$ is the

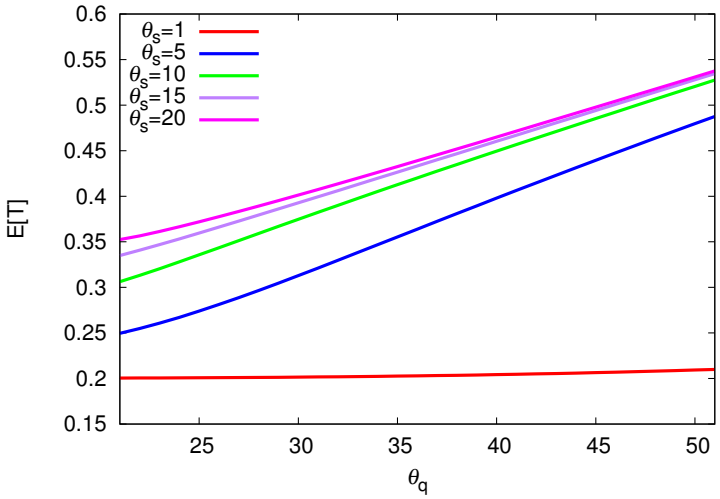


Figure 3.2.: The mean job-response time E[T] vs. θ_q.

case in which the system is always in normal-operation mode. We set $\zeta = 0.5$ ($\mu_S = 2.5, \mu_F = 5$) and $\lambda = 50$.

We observe from Figure 3.2 that when θ_q increases, E[T]'s for all the cases except $\theta_s = 1$ grow linearly. In Figure 3.3, on the other hand, E's for all the cases except $\theta_s = 1$ decrease monotonically. Note that a large θ_q makes the power-saving period large. This results in a large job-response time and a small energy consumption.

In terms of θ_s, we observe that E[T] becomes large with the increase in θ_s, while E decreases significantly. This is because a large θ_s makes the power-saving period large, resulting in a large job-response time and small energy consumption. From both figures, we observe that there exist a trade-off relation between E[T] and E.

Figure 3.4 shows how E[T] is affected by θ_s. We plot E[T]'s in cases of $\theta_q = 25$, 30, 35, and 40. In this figure, we set $\zeta = 0.5$ ($\mu_S = 2.5$) and $\lambda = 50$, as in Figures 3.2 and 3.3. We observe from Figure 3.4 that E[T] grows with the increase in θ_s. This is because the system is likely to be in power-saving mode when θ_s is large. We also observe that E[T] for $\theta_q = 25$ is the smallest among the four cases of θ_q, and that E[T] for a small θ_q grows more gradually than that for a large θ_q. In particular, the case of $\theta_q = 25$ shows the slowest increase against θ_s.

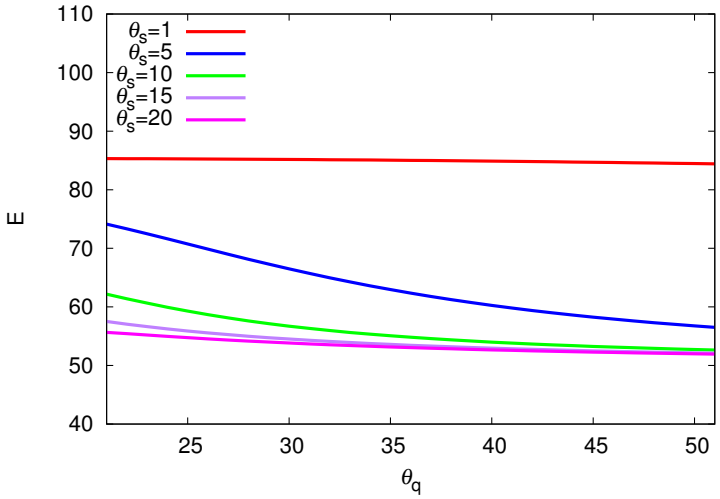


Figure 3.3.: The mean energy consumption E vs. θ_q.

Figure 3.5 shows the mean energy consumption E against θ_s. We also set $\zeta = 0.5$ ($\mu_S = 2.5$) and $\lambda = 50$, and we plot E's in cases of $\theta_q = 25$, 30, 35, and 40. We observe from this figure that E decreases monotonically with the increase in θ_s, as expected. We also observe that E for $\theta_q = 25$ is the largest among the four cases of θ_q.

Note that in Figures 3.4 and 3.5, both the job-response time and mean power consumption gradually change with the increase in θ_s when $\theta_q = 25$. This feature is significant because we can widely control the job-response time and power consumption by adjusting θ_s under setting of a small θ_q.

3.3.2. Impact of Power-saving Level on Performance

In this subsection, we investigate how the power-saving level ζ affects system performance.

Figure 3.6 represents the mean job-response time E[T] against the power-saving level ζ. In this figure, we set $\lambda = 50$ and $\theta_q = 35$. We observe from Figure 3.6 that E[T] monotonically decreases with the increase in ζ for $\theta_s = 15$ and 10. Note that ζ is the ratio of the service rate of power-saving mode to that of normal-operation mode. Therefore, a large ζ implies that the service rate of power-saving mode is close to that of normal-operation mode. In other words, the overall service rate

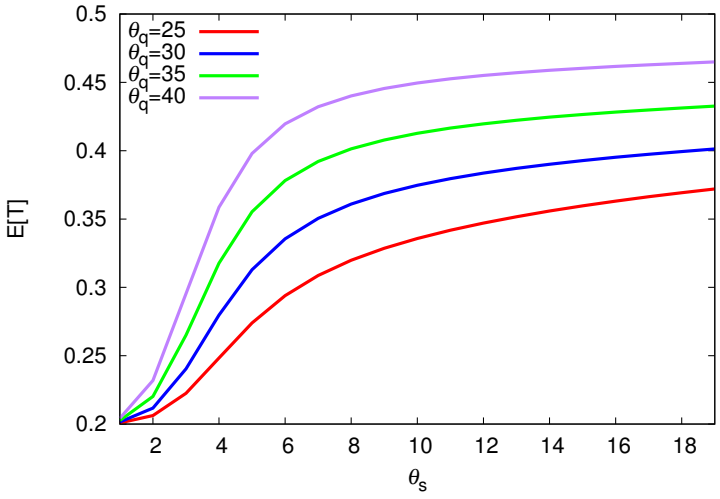


Figure 3.4.: The mean job-response time E[T] vs. θ_s.

of the system is large for a large ζ, resulting in a small job-response time.

In Figure 3.6, E[T] for $\theta_s = 5$ rapidly decreases, then gradually increases and finally decreases along the E[T]'s for the other two cases of θ_s. Remind that θ_s is the threshold for the system-state transition from normal-operation mode to power-saving one. When the number of jobs in the system is smaller than θ_s, power-saving mode starts. The time interval from this point to the ending point of power-saving mode (the initiation point of normal-operation mode) for a small θ_s is likely to be longer than that for a large θ_s under a fixed θ_q. This implies that the number of jobs served in this time interval becomes large for a small θ_s, resulting in a large job-response time.

The above explanation is confirmed from Figure 3.7. Figure 3.7 shows the mean power consumption E against ζ. In this figure, we set $\lambda = 50$ and $\theta_q = 35$, as is the case of Figure 3.6. In this figure, E for $\theta_s = 15$ grows monotonically with the increase in ζ. On the other hand, E's for $\theta_s = 10$ and 5 increase first, then decrease, and finally increases along the case of $\theta_s = 15$. As we explained the above paragraph, the period of power-saving mode is likely to be long, resulting in a small power consumption for job processing.

These results imply that saving a large amount of energy at power-saving mode does not always reduce the overall power consumption. Figures 3.6 and 3.7 sug-

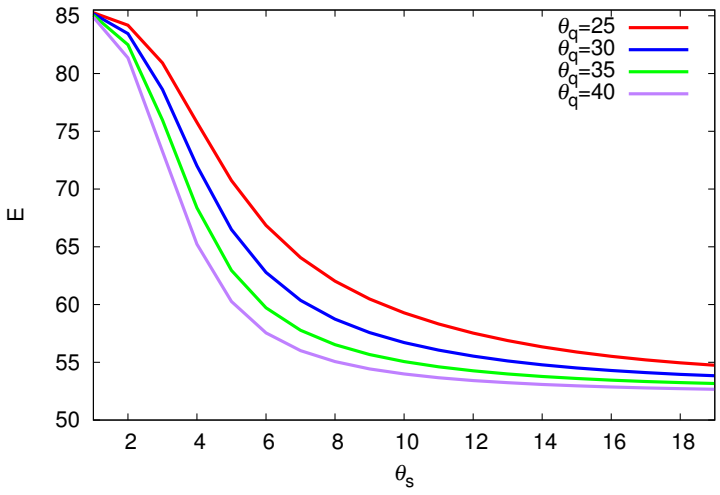


Figure 3.5.: The mean energy consumption E vs. θ_s.

gests that $\zeta = 0.5$ achieves a better performance of energy consumption for all the five θ_s cases, providing a moderate job-response time. $\zeta = 0.5$ is the case where a half of servers in a cluster are turned off at power-saving mode, and this also confirms our finding in Chapter 2.

3.3.3. Impact of Job Arrival Rate on Performance

In this subsection, we investigate how the job arrival rate affects system performance. We set $\zeta = 0.5$.

Figures 3.8 to 3.10 show the mean job-response time E[T] against θ_q. In each figure, we plot five cases of $\theta_s = 1$, 5, 10, 15 and 20. The curves of M/M/c's with service rates μ_S and μ_F are also depicted for reference. Figure 3.8 shows the case of $\lambda = 20$. Note that we have $\lambda\,(= 20) \ll c\mu_S \leq c\mu_F$. Since the number of jobs in the system more than θ_q rarely occurs, the service rate of most jobs is μ_S. This results in that E[T] stays constant.

Figure 3.9 shows E[T] for $\lambda = 40$, and we have $\lambda\,(= 40) < c\mu_S \leq c\mu_F$. When $\lambda = 40$, some jobs are likely to be served with rate μ_F since the number of jobs reaches θ_q frequently. We also observe that when θ_q increases, E[T]'s for all θ_s grow monotonically and converge to the job-response time with rate μ_S.

Figure 3.10 is the case of $\lambda = 50$, and $c\mu_S \leq \lambda\,(= 50) < c\mu_F$ holds. Note that in

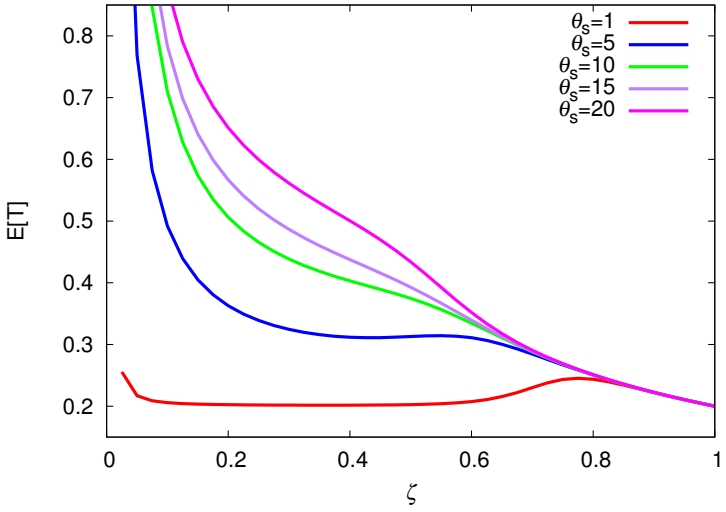


Figure 3.6.: The mean job-response time E[T] vs. ζ.

this case, M/M/c with service rate μ_S is unstable and hence we cannot calculate $E[T]$. We observe in this figure that E[T]'s for all the cases except $\theta_s = 1$ grow linearly with the increase in θ_q.

Figures 3.11 to 3.13 show ERP against θ_q. In each figures, we plot five curves of $\theta_s = 1$, 5, 10, 15 and 20. We also plot the case of M/M/c with service rate μ_F as reference. Figure 3.11 shows the case of $\lambda = 20$, and we have $\lambda\,(= 20) \ll c\mu_S \leq c\mu_F$. Since the mean job-response time E[T] and the mean amount of energy consumption E stay constant, ERP is constant. Figure 3.12 shows ERP for $\lambda = 40$, and we have $\lambda\,(= 40) < c\mu_S \leq c\mu_F$. As is the case with E[$T$], we observe that ERP's for all θ_s grow monotonically and converges to ERP with rate μ_S. Figure 3.13 is the case of $\lambda = 50$, and $c\mu_S \leq \lambda\,(= 50) < c\mu_F$ holds. We observe that ERP's for all the cases increase monotonically with the increase in θ_q.

These result imply that in order to minimize ERP, reducing both θ_s and θ_q is effective in most cases. Note that reducing both θ_s and θ_q does not always provide a small ERP. Figures 3.14 and 3.15 show ERP against λ. These figures show that ERP for small $\theta_s\,(= 1)$ and small $\theta_q\,(= 21)$ is greater than that for M/M/c with rate μ_S when λ is small (around 25).

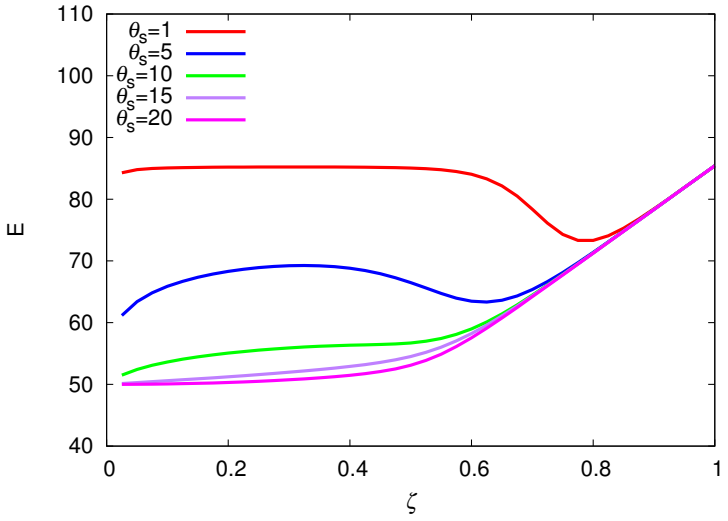


Figure 3.7.: The mean energy consumption E vs. ζ.

3.4. Conclusion

In this chapter, we modeled a data center as a multiple-server queue with job service depending on the number of jobs in the system.

We found that when the threshold for switching power-saving mode to normal-operation one is small, the job-response time and power consumption changes gradually. We also investigated how those performance measures are affected by power-saving level of a cluster. We confirmed from the numerical results that turning a half of servers in a cluster off is effective both for keeping the mean job-response time small and for saving energy consumption. Numerical example also showed that reducing both θ_s and θ_q is effective in order to minimize ERP in most cases.

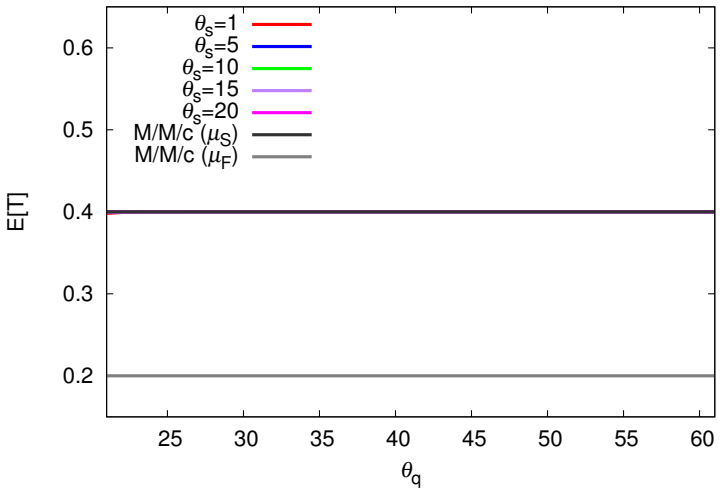


Figure 3.8.: The mean job-response time E[T] vs. θ_q with $\lambda = 20$.

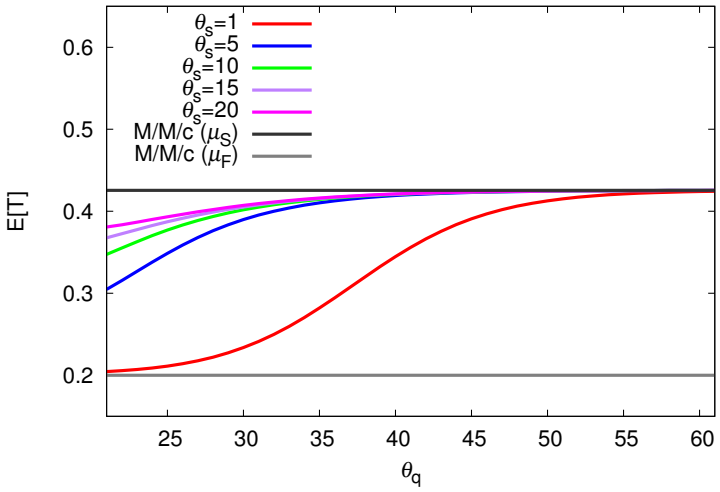


Figure 3.9.: The mean job-response time E[T] vs. θ_q with $\lambda = 40$.

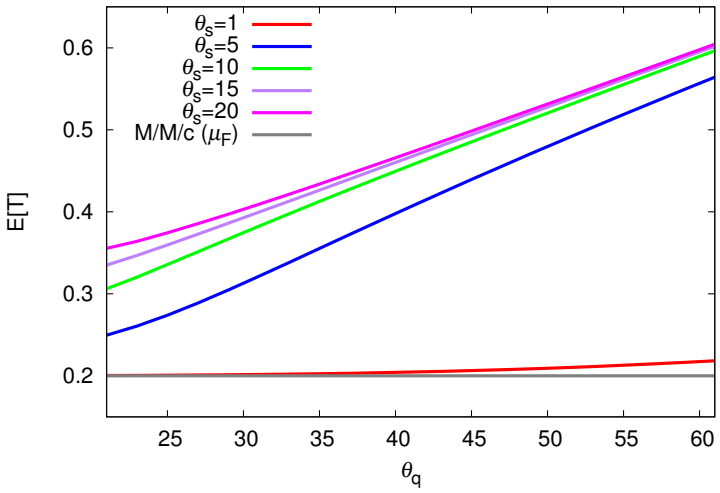


Figure 3.10.: The mean job-response time E[T] vs. θ_q with $\lambda = 50$.

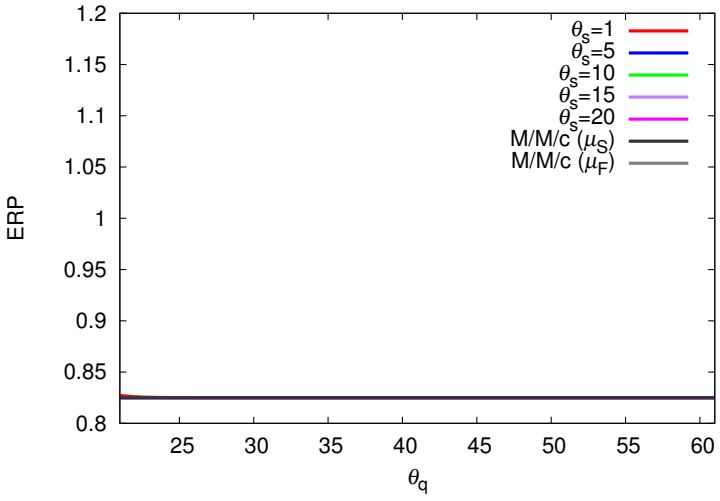


Figure 3.11.: The Energy-Response time Product vs. θ_q with $\lambda = 20$.

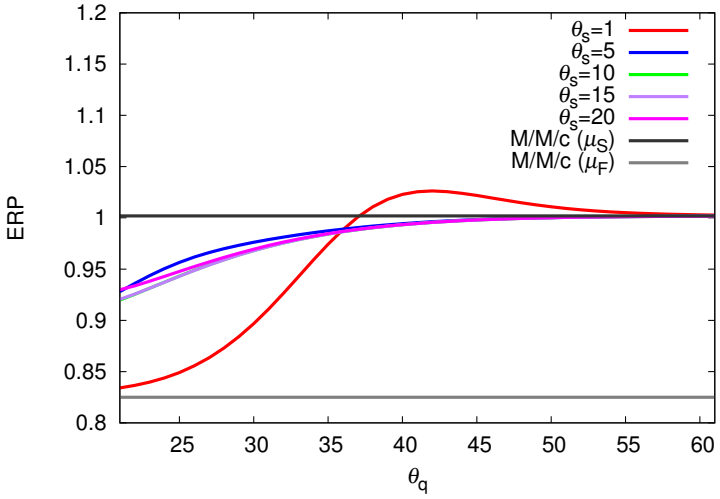


Figure 3.12.: The Energy-Response time Product vs. θ_q with $\lambda = 40$.

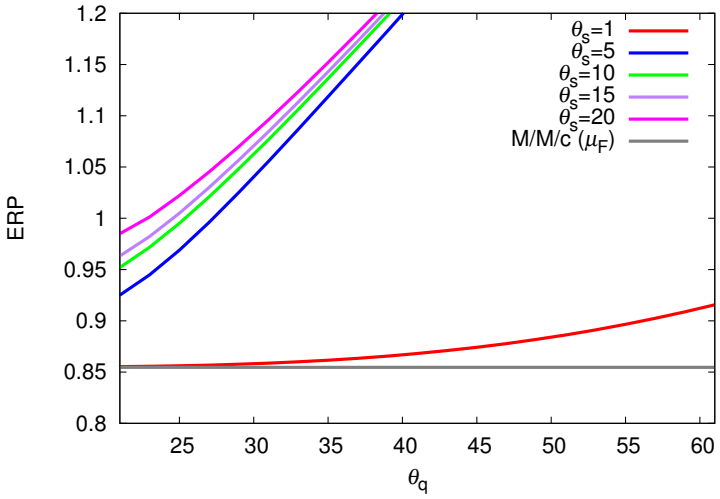


Figure 3.13.: The Energy-Response time Product vs. θ_q with $\lambda = 50$.

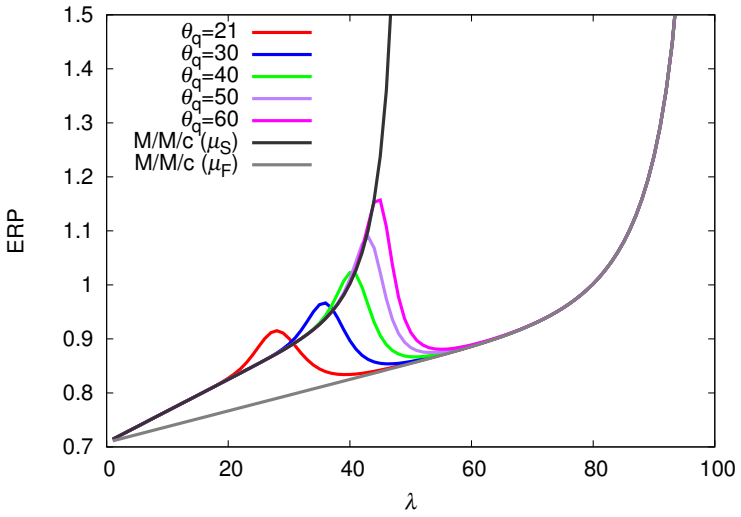


Figure 3.14.: The Energy-Response time Product vs. λ with $\theta_s = 1$.

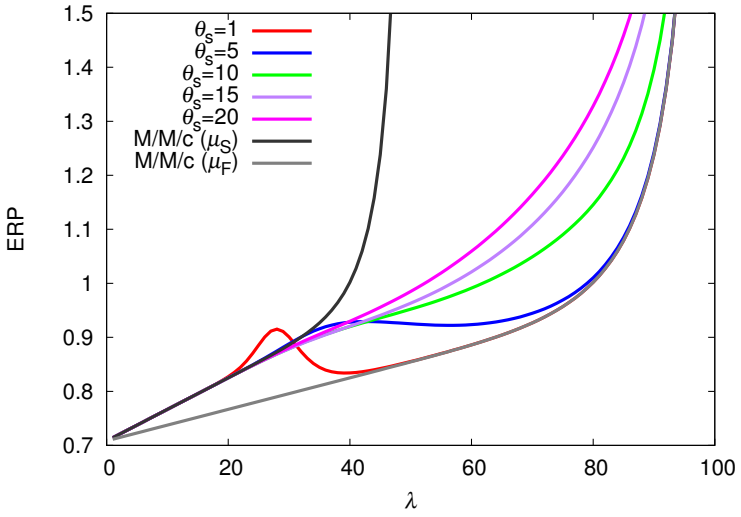


Figure 3.15.: The Energy-Response time Product vs. λ with $\theta_q = 21$.

4. Conclusion

In this thesis, we considered a power-management scheme for data centers with server clusters. We focused on a data center accommodating a large number of server clusters, each of which consisted of several server machines, providing parallel distributed computing service. In order to investigate the performance of the cluster-based power management scheme, for data centers, we considered two multi-server queueing systems. In the first system, the power-operation modes alternated according to the state of a background process. In the second system, on the other hand, we considered the power management for which the power-operation mode depends on the number of jobs in the system.

In the first system, we considered a queueing model for data centers with BEEMR-like energy-saving management mechanism. We modeled a data center as a multiple-server queue with job service depending on a background process. Using matrix-analytic method, we derived the joint distribution of the number of jobs in the system and the state of the background process, yielding the mean job-response time and mean amount of energy consumption as performance measures. Numerical examples showed that the amount of energy consumption grows linearly with the increase in the number of clusters. We also confirmed that turning a half of servers in a cluster off is effective for keeping the mean job-response time small.

In the second system, we modeled a data center as a multiple-server queue with job service depending on the number of jobs in the system. We found that when the threshold for switching power-saving mode to normal-operation one is small, the job-response time and power consumption changes gradually. We also investigated how those performance measures are affected by power-saving level of a cluster. We found that we can widely control the job-response time and power consumption by adjusting θ_s under setting of a small θ_q. It was also shown that

turning a half of servers in a cluster off provides not only a better performance of energy consumption, but also a moderate job-response time. Furthermore, in order to minimize ERP, reducing both θ_s and θ_q is effective in most cases.

In this thesis, we considered that the job-processing times in a cluster are exponentially distributed for both normal and energy-saving modes. In real data centers, however, the job-processing times follows some heavy-tailed distribution. In order to consider this heavy-tailedness of the job-processing time, we need to consider a multi-server queueing model with a general service time. In general, M/G/c queue is an open problem and hence we have to develop some approximation techniques for investigating the tradeoff between the job-response time and energy consumption. This is our future work.

A. Stability Condition

When $c = 2$,we calculate (3.2). Matrix $\boldsymbol{A} = \boldsymbol{A}_0 + \boldsymbol{A}_1 + \boldsymbol{A}_2$ is given by

$$\boldsymbol{A} = \begin{bmatrix} -\alpha_S - 2\mu_F & \alpha_S & 2\mu_F & 0 & 0 & 0 \\ \alpha_F & -\alpha_F & 0 & 0 & 0 & 0 \\ 0 & 0 & -\alpha_S - \mu_F & \alpha_S & \mu_F & 0 \\ 0 & \mu_S & \alpha_F & -\alpha_F - \mu_S & 0 & 0 \\ 0 & 0 & 0 & 0 & -\alpha_S & \alpha_S \\ 0 & 0 & 0 & 2\mu_S & \alpha_F & -\alpha_F - 2\mu_S \end{bmatrix}.$$

By solving the system of equations with six variables, $\boldsymbol{\pi}_A$ satisfying (2.2) is denoted by

$$\boldsymbol{\pi}_A^T = \frac{1}{K_2} \begin{bmatrix} \alpha_S\alpha_F\mu_S^2(\alpha_S + \mu_F) \\ \alpha_S\mu_S^2(\alpha_S + \mu_F)(\alpha_S + 2\mu_F) \\ 2\alpha_S\alpha_F\mu_S\mu_F(\alpha_F + \mu_S) \\ 2\alpha_S\alpha_F\mu_S\mu_F(\alpha_S + \mu_F) \\ \alpha_F\mu_F^2(\alpha_F + \mu_S)(\alpha_F + 2\mu_S) \\ \alpha_S\alpha_F\mu_F^2(\alpha_F + \mu_S), \end{bmatrix},$$

where

$$K_2 = (\alpha_S + \alpha_F)(\alpha_S\mu_S + \alpha_F\mu_F + \mu_S\mu_F)(\alpha_S\mu_S + \alpha_F\mu_F + 2\mu_S\mu_F).$$

Therefore,

$$\begin{aligned} &\boldsymbol{\pi}_A\boldsymbol{A}_0\boldsymbol{e}_1 = \lambda, \\ &\boldsymbol{\pi}_A\boldsymbol{A}_2\boldsymbol{e}_1 = \frac{2\mu_S\mu_F\left(\alpha_S^2 + 2\alpha_S\alpha_F + \alpha_F^2 + \alpha_S\mu_F + \alpha_F\mu_S\right)}{(\alpha_S + \alpha_F)(\alpha_S\mu_S + \alpha_F\mu_F + \mu_S\mu_F)}. \end{aligned}$$

Similarly, we consider the case of $c = 3$.

$$[A]_{ij} = \begin{cases} \alpha_S & i = 1, j = 2, \text{or } i = 3, j = 4, \text{or } i = 5, j = 6, \text{or } i = 7, j = 8, \\ -\alpha_S & i = 7, j = 7, \\ \mu_F, & i = 5, j = 7, \\ -\alpha_S - \mu_F, & i = 5, j = 5, \\ 2\mu_F, & i = 3, j = 5, \\ -\alpha_S - 2\mu_F, & i = 3, j = 3, \\ 3\mu_F, & i = 1, j = 3, \\ -\alpha_S - 3\mu_F, & i = 1, j = 1, \\ \alpha_F & i = 2, j = 1, \text{or } i = 4, j = 3, \text{or } i = 6, j = 5, \text{or } i = 8, j = 7, \\ -\alpha_F & i = 2, j = 2, \\ \mu_S & i = 4, j = 2, \\ -\alpha_F - \mu_S, & i = 4, j = 4, \\ 2\mu_S & i = 6, j = 4, \\ -\alpha_F - 2\mu_S, & i = 6, j = 6, \\ 3\mu_S & i = 8, j = 6, \\ -\alpha_F - 3\mu_S, & i = 8, j = 8. \\ 0 & \text{otherwise.} \end{cases}$$

We solve the system of equations with eight variables in (2.2)

$$\boldsymbol{\pi}_A^T = \frac{1}{K_3} \begin{bmatrix} \alpha_S\alpha_F\mu_S^3(\alpha_S + \mu_F)(\alpha_S + 2\mu_F) \\ \alpha_S\mu_S^3(\alpha_S + \mu_F)(\alpha_S + 2\mu_F)(\alpha_S + 3\mu_F) \\ 3\alpha_S\alpha_F\mu_S^2\mu_F(\alpha_S + \mu_F)(\alpha_F + \mu_S) \\ 3\alpha_S\alpha_F\mu_S^2\mu_F(\alpha_S + \mu_F)(\alpha_S + 2\mu_F) \\ 3\alpha_S\alpha_F\mu_S\mu_F^2(\alpha_F + \mu_S)(\alpha_F + 2\mu_S) \\ 3\alpha_S\alpha_F\mu_S\mu_F^2(\alpha_S + \mu_F)(\alpha_F + \mu_S) \\ \alpha_F\mu_F^3(\alpha_F + \mu_S)(\alpha_F + 2\mu_S)(\alpha_F + 3\mu_S) \\ \alpha_S\alpha_F\mu_F^3(\alpha_F + \mu_S)(\alpha_F + 2\mu_S) \end{bmatrix},$$

where

$$\begin{aligned} K_3 =&(\alpha_S + \alpha_F)(\alpha_S\mu_S + \alpha_F\mu_F + \mu_S\mu_F) \\ &(\alpha_S\mu_S + \alpha_F\mu_F + 2\mu_S\mu_F)(\alpha_S\mu_S + \alpha_F\mu_F + 3\mu_S\mu_F). \end{aligned}$$

Therefore,

$$\begin{aligned} \boldsymbol{\pi}_A \boldsymbol{A}_0 \boldsymbol{e}_1 &= \lambda, \\ \boldsymbol{\pi}_A \boldsymbol{A}_2 \boldsymbol{e}_1 &= \frac{3\mu_S\mu_F\left(\alpha_S^2 + 2\alpha_S\alpha_F + \alpha_F^2 + \alpha_S\mu_F + \alpha_F\mu_S\right)}{(\alpha_S + \alpha_F)(\alpha_S\mu_S + \alpha_F\mu_F + \mu_S\mu_F)}. \end{aligned}$$

As stated above, for all $c \geq 2$, we conjecture

$$\begin{aligned} \boldsymbol{\pi}_A \boldsymbol{A}_0 \boldsymbol{e}_1 &= \lambda, \\ \boldsymbol{\pi}_A \boldsymbol{A}_2 \boldsymbol{e}_1 &= \frac{c\mu_S\mu_F\left(\alpha_S^2 + 2\alpha_S\alpha_F + \alpha_F^2 + \alpha_S\mu_F + \alpha_F\mu_S\right)}{(\alpha_S + \alpha_F)(\alpha_S\mu_S + \alpha_F\mu_F + \mu_S\mu_F)}. \end{aligned}$$

Acknowledgements

I would like to thank the following people for their support.

I would like to express my gratitude to Professor Shoji Kasahara for his support. His guidance helped me during my research and writing of this thesis. Without his help, this thesis would not have been possible.

Co-supervisor Kazushi Ikeda's questions and comments were invaluable when I made my Master's 2nd presentation.

I would like to thank co-supervisor Masahiro Sasabe and Assistant Professor Jun Kawahara. They gave me insightful suggestions throughout two years of graduate school.

I would like to express the deepest appreciation to my fellow labmates, who gave me warm encouragement.

Lastly, I also wish to thank my parents for their financial support.

References

[1] L. A. Barroso and U. Hölzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan & Claypool, 2009.

[2] R. Brown, E. Masanet, B. Nordman, W. Tschudi, A. Shehabi, J. Stanley, J. Koomey, D. Sartor, P. Chan, J. Loper, S. Capana, B. Hedman, R. Duff, E. Haines, D. Sass and A. Fanara, *Report to congress on server and data center energy efficiency: Public law 109–431*, Lawrence Berkeley National Laboratory, LBNL-363E, 2007.

[3] Y. Chen, S. Alspaugh, D. Borthakur and R. Katz, Energy Efficiency for Large-Scale MapReduce Workloads with Significant Interactive Analysis, *Proc. The European Professional Society on Computer Systems 2012*, (2012), 43-56.

[4] A. Gandhi, V. Gupta, M. Harchol-Balter and M. Kozuch, Optimality Analysis of Energy-Performance Trade-Off for Server Farm Management, *Performance Evaluation*, **67** (2010), 1155–1171.

[5] M. Kato, H. Masuyama, S. Kasahara, and Y. Takahashi, Effect of Energy-Saving Server Scheduling on Power Consumption for Large-Scale Data Centers, *Journal of Industrial and Management Optimization*, **12** (2016), 667–685.

[6] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, ASA-SIAM, 1999.

[7] R. Nelson, *Probability, Stochastic Processes, and Queueing Theory*, Springer Verlag, 2000.

[8] S. Pelley, D. Meisner, T. F. Wenisch and J. W. VanGilder, Understanding and abstracting total data center power, *Proc. Workshop on Energy-Efficient Design 2009*, Jun. 2009.

[9] T. Sakata, and S. Kasahara, Multi-server Queue with Job Service Time Depending on a Background Process, *Proc. The 10th International Conference on Queueing Theory and Network Applications* (eds. T. V. Do, Y. Takahashi, W. Yue, and V.-H. Nguyen), **383** (2015), 163–171.

[10] C. Schwarts, R. Pries and P. Tran-Gia, A Queuing Analysis of an Energy-Saving Mechanism in Data Centers, *Proc. International Conference on Information Networking 2012*, (2012), 70–75.

Publication List

1. Conference paper

   [1] T. Sakata, and S. Kasahara, Multi-server Queue with Job Service Time Depending on a Background Process, *Proc. The 10th International Conference on Queueing Theory and Network Applications* (eds. T. V. Do, Y. Takahashi, W. Yue, and V.-H. Nguyen), **383** (2015), 163–171.

2. Domestic Conferences

   [1] 佐嘉田智之, 笠原正治, “サービス受付時の背後過程に依存したサービス時間をもつ複数サーバ待ち行列,” 2014 年度確率モデルシンポジウム, (2015), 122–131.

   [2] 佐嘉田智之, 笠原正治, “系内ジョブ数に依存した電源管理法をもつデータセンター待ち行列モデル,” 日本オペレーションズ・リサーチ学会 2015 年度秋季研究発表会アブストラクト集, (2015), 160–161.

   [3] 佐嘉田智之, 笠原正治, “Performance Analysis of Power Management Scheme for Data Centers,” 2015 年度待ち行列シンポジウム, (2016), 35–44.