


Optimal load distribution for multiple classes of applications on heterogeneous servers with variable speeds

Keqin Li 

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

Correspondence

Keqin Li, Department of Computer Science, State University of New York, New Paltz, New York, NY 12561, USA.
Email: lik@newpaltz.edu

Summary

Performance and power are 2 significant issues in cloud computing. It is a critical issue on how to provide the best quality of service by consuming certain available power resource. For a given application environment and a given group of servers, optimal load distribution and optimal server speed setting can be an effective way to deal with the power-performance tradeoff. The technique of variable and task-type-dependent server speed management can be explored to optimize the server performance and to minimize the power consumption of a server with mixed applications. In this paper, we consider the problem of optimal load distribution for multiple classes of applications on heterogeneous servers with variable speeds. Given several classes of applications characterized by their arrival rates and expected execution requirements, several heterogeneous servers characterized by their power consumption parameters, and certain power supply, our problem is formulated as a multivariable optimization problem, ie, finding an optimal load distribution and an optimal server speed setting, such that the average task response time is minimized. To study the problem analytically, each server is treated as an M/G/1 queueing system with mixed classes of tasks such that both the average response time and the average power consumption can be calculated analytically. We define a power constrained performance optimization problem and develop a numerical algorithm to solve our optimization problem by solving a system of nonlinear equations. We also demonstrate numerical examples to show the effectiveness of our model and method. To the best of our knowledge, such analytical study of optimal load distribution and optimal server speed setting for multiple classes of applications on heterogeneous servers with variable speeds has not been available in the existing literature.

KEYWORDS

average response time, heterogeneous server with variable speed, multiple classes of applications, optimal load distribution, optimal server speed setting, power consumption, task-type-dependent server speed management

1 | INTRODUCTION

Performance and power are 2 significant issues in cloud computing. From a cloud consumer's point of view, quality of service (QoS) is an important concern, which is a key factor in satisfying a user's experience and expectation and in

choosing a service provider. While there are many different perspectives of QoS for different applications, the average response time is a commonly adopted performance metric. From a service provider's point of view, cost of service is an important concern. Contemporary warehouse-scale data centers consume significant amount of energy. The power consumed by IT equipment plus the overhead power consumed in power delivery and cooling can be over 30% of the overall operating expenses and a significant portion of the total cost of ownership of a data center.¹ Therefore, it is a critical issue on how to provide the best QoS by consuming certain available power resource.

A data center is a massive collection of servers that provide computation, storage, and communication services. It has been pointed out that future scaling of data center capability depends upon improvements to server power efficiency. Most of the future opportunity to improve data center power efficiency lies in improving the power efficiency of the servers themselves, as most of the inefficiency in the rest of a data center has largely been eliminated.² Thus, an important question for data center operators is how to balance the workload among the servers and how to decide the server capacity (number of servers and speeds of servers) so as to minimize the average task response time without exceeding certain power limitation. For a given application environment and a given group of servers, optimal load distribution and optimal server speed setting can be an effective way to deal with the power-performance tradeoff.

The technique of workload-dependent dynamic power management refers to dynamic power and speed adjustment according to the current workload.³ It is a powerful way of fine server tuning for applications with different characteristics. The technique of variable and task-type-dependent server speed management can be explored to optimize the server performance and to minimize the power consumption of a server with mixed applications. For instance, the power supply and the server speed can be increased for a type of applications with greater arrival rate and greater coefficient of variation of execution requirement.⁴ Such runtime power and speed adjustment can be supported by a mechanism called dynamic voltage scaling, or equivalently, dynamic frequency scaling, dynamic speed scaling, or dynamic power scaling.⁵

In this paper, we consider the problem of optimal load distribution for multiple classes of applications on heterogeneous servers with variable speeds. There are multiple classes of applications with different arrival rates and execution requirements. The heterogeneous servers have different speeds in processing different classes of applications. Furthermore, they have different power consumption parameters. Given several classes of applications characterized by their arrival rates and expected execution requirements, several heterogeneous servers characterized by their power consumption parameters, and certain power supply, our problem is formulated as a multivariable optimization problem, ie, finding an optimal load distribution and an optimal server speed setting, such that the average task response time is minimized. The main contributions of the paper are as follows.

- To study the problem analytically, each server is treated as an M/G/1 queueing system with mixed classes of tasks such that both the average response time and the average power consumption can be calculated analytically.
- We define a power constrained performance optimization problem and develop a numerical algorithm to solve our optimization problem by solving a system of nonlinear equations.
- We also demonstrate numerical examples to show the effectiveness of our model and method.

To the best of our knowledge, such analytical study of optimal load distribution and optimal server speed setting for multiple classes of applications on heterogeneous servers with variable speeds has not been available in the existing literature. The proposed problem and algorithm can be applied to practical cloud computing systems with multiple classes of applications.

The rest of the paper is organized as follows. In Section 2, we review related research. In Section 3, we present our queueing model for heterogeneous servers with mixed applications. In Section 4, we formulate our power constrained performance optimization problem. In Section 5, we develop our numerical algorithm. In Section 6, we demonstrate a numerical example. In Section 7, we conclude the paper.

2 | RELATED RESEARCH

Data center power efficiency has been studied by many researchers. Gandhi investigated new approaches to dynamic server provisioning to increase server utilization and to reduce data center power consumption.⁶ Ganesh et al proposed an integrated approach, which combines the benefits of the power proportional approach (focusing on reducing disk and server power consumption) and the green data center approach (focusing on reducing power consumed by support infrastructure like cooling equipment, power distribution units, and power backup equipment).⁷ Leverich explored 4 compelling opportunities to improve server power efficiency, ie, 2 hardware proposals that explicitly reduce the power

consumption of servers and 2 software proposals that improve the power efficiency of servers operating as a cluster.² Pakbaznia and Pedram addressed server consolidation concurrently with task assignment by formulating the resulting optimization problem as an integer linear programming problem and solving the problem by using a heuristic algorithm in polynomial time.⁸ Tuncer et al presented a data center power budgeting policy that simultaneously improves the QoS and power efficiency by considering the workload and cooling induced asymmetries among the servers.⁹ Zapater et al developed empirical models to estimate the contributions of static and dynamic power consumption in enterprise servers for a wide range of workloads and analyzed the interactions between temperature, leakage, and cooling power for various workload allocation policies.¹⁰

Several surveys and comparative studies have been conducted for the extensive research in cloud load balancing and load distribution. Al Sallami discussed and compared existing load balancing techniques in cloud computing based on various parameters.¹¹ Himanshi and Ahuja explored autonomic approaches for optimizing provisioning for heterogeneous workloads on enterprise grids and clouds, and reviewed load balancing strategies for cloud infrastructures.¹² Kapoor surveyed various dynamic load balancing algorithms in cloud with discussion and comparison of the pros and cons of these algorithms.¹³ Katyal and Mishra presented a comparative study of various load balancing schemes in different cloud environments based on requirements specified in service level agreement.¹⁴ Kaur and Luthra gave an overview of many load balancing algorithms that help to achieve better throughput and improve the response time in cloud environments.¹⁵ Khiyaita et al gave an overview of load balancing in cloud computing by exposing the most important research challenges.¹⁶ Al Nuaimi et al 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.¹⁷ Rahman et al provided a comprehensive review on the existing load balancing strategies and presented load balancer as a service model adopted by the major market players.¹⁸ Shameem and Shaji presented a survey of dynamic load balancing strategies on cloud with the focus on various metrics to analyze the efficacy of the existing techniques.¹⁹ Singh et al compared various load balancing algorithms on the basis of their metrics.²⁰

Numerous researchers have investigated various approaches to cloud load balancing. Anjali et al showed a new approach to dynamic load balancing using the concept of mobile agent, ie, a software program that executes independently and performs the basic task.²¹ Dasgupta et al 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.²² Dhinesh and Krishna 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.²³ Gasior and Seredynski 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 2-dimensional cellular automata space.²⁴ Gopinath and Vasudevan focused on 2 load balancing algorithms in cloud, ie, Min-Min and Max-Min, to minimize the response time and the waiting time.²⁵ Grover and Katiyar used an agent-based dynamic load balancing approach that greatly reduces the communication cost of servers, accelerates the rate of load balancing, and improves the throughput and the response time of the cloud.²⁶ Li studied the problem of optimal distribution of generic tasks over 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.²⁷ Liu et al took a game approach to multiservers load balancing with load-dependent server availability consideration.²⁸ Sahu et al 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.²⁹ Singh et al proposed an autonomous agent-based load balancing algorithm, which provides dynamic load balancing for cloud environment.³⁰ Srinivasan et al used an enhanced shortest job-first scheduling algorithm to achieve reduced response time and reduced starvation and job rejection rate.³¹ Tong et al developed an approach from machine learning to learn task arrival and execution patterns online, ie, automatically acquiring such knowledge without any beforehand modeling and proactively allocating tasks on account of the forthcoming tasks and their execution dynamics.³² Xiao et al studied the collaboration among benevolent clouds that are cooperative in nature and willing to accept jobs from other clouds, took advantage of machine learning, and proposed a distributed scheduling mechanism to learn the knowledge of job model, resource performance, and others' policies.³³ Xiao et al also 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.³⁴

Cloud load distribution has been considered together with energy consumption. Beloglazov et al conducted a survey of research in energy-efficient computing and proposed architectural principles for the energy-efficient management of clouds and energy-efficient resource allocation policies and scheduling algorithms considering QoS expectations and power usage characteristics of the devices.³⁵ Cao et al addressed optimal power allocation and load distribution

for multiple heterogeneous multicore server processors across clouds and data centers as optimization problems.³⁶ Ghafari et al proposed a new power-aware load balancing algorithm based on artificial bee colony to detect both overutilized and underutilized hosts for effective power management.³⁷ Huang et al studied the problem of power consumption minimization with performance constraint in heterogeneous distributed embedded systems by optimal load distribution.³⁸ Kansal and Chana 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.³⁹ Li addressed the issue of optimal task dispatching on multiple heterogeneous multiserver systems with dynamic speed and power management.⁴⁰ Malik et al 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.⁴¹ Paul et al investigated load distribution strategies to minimize the electricity cost and increase renewable energy integration subject to compliance with service level agreement with consideration of the adverse effects of switching the servers.⁴² Tian et al investigated performance and power tradeoff for multiple heterogeneous servers by considering 2 problems, ie, optimal job scheduling with fixed service rates and joint optimal service speed scaling and job scheduling.⁴³ Yang et al employed a game theoretic approach to solve the problem of minimizing energy consumption as a Stackelberg game and modeled

TABLE 1 Summary of the notations used in the paper

Notation	Definition
m	Number of classes of applications
n	Number of heterogeneous servers
$\tilde{\lambda}_i$	The arrival rate of the i th type of applications
λ	The total task arrival rate, ie, $\tilde{\lambda}_1 + \tilde{\lambda}_2 + \dots + \tilde{\lambda}_m$
$\lambda_{i,j}$	The arrival rate of the substream of tasks assigned to server j
λ_j	The total task arrival rate to server j , ie, $\lambda_{1,j} + \lambda_{2,j} + \dots + \lambda_{m,j}$
r_i	The execution requirements of the tasks of the i th type of applications
\bar{r}_i, r_i^2	Mean and second moment of r_i
$s_{i,j}$	The execution speed for the i th type of applications on server j
$x_{i,j}$	The execution times of the tasks of the i th type of applications on server j
$\bar{x}_{i,j}, x_{i,j}^2$	Mean and second moment of $x_{i,j}$
x_j	The execution time of a task on server j
\bar{x}_j, x_j^2	Mean and second moment of x_j
ρ_j	The utilization of server j
W_j	The average waiting time of a task on server j
σ_j	$\lambda_j \bar{x}_j^2 = \lambda_{1,j} \bar{x}_{1,j}^2 + \lambda_{2,j} \bar{x}_{2,j}^2 + \dots + \lambda_{m,j} \bar{x}_{m,j}^2$
$T_{i,j}$	The average response time of the tasks of the i th type of applications on server j
T_j	The average response time of all tasks on server j
T	The average response time of all tasks on the n servers
P_j^*	Base power consumption of server j
ξ_j, α_j	Parameters of the dynamic power consumption of server j
P_j	The average power consumption of server j
P	The total power consumption of the n servers
\tilde{P}	Power constraint
L_i	$\lambda_{i,1} + \lambda_{i,2} + \dots + \lambda_{i,n}$
ϕ_i, ψ	Lagrange multipliers
N	$2mn + m + 1$
\mathbf{y}	$(y_1, y_2, \dots, y_N) = (\lambda_{1,1}, \dots, \lambda_{m,n}, s_{1,1}, \dots, s_{m,n}, \phi_1, \dots, \phi_m, \psi)$
$G_{i,j}, H_{i,j}, J_i, K$	$2mn + m + 1$ nonlinear equations
F_k	$F_{(i-1)n+j} = G_{i,j}, F_{mn+(i-1)n+j} = H_{i,j}, F_{2mn+i} = J_i, F_N = K, 1 \leq i \leq m, 1 \leq j \leq n$
$\mathbf{F}(\mathbf{y})$	$(F_1(\mathbf{y}), F_2(\mathbf{y}), \dots, F_N(\mathbf{y}))$
$\mathbf{J}(\mathbf{y})$	The Jacobian matrix of $\mathbf{F}(\mathbf{y})$

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.⁴⁴

3 | THE MODEL

Throughout the paper, we use \bar{x} to represent the expectation of a random variable x . Table 1 provides a summary of the notations used in this paper.

There are m classes of applications. Each class of applications is characterized by its arrival rate and execution requirement. We consider n heterogeneous servers with variable execution speeds. Each server has its own speed in processing a class of applications and its own power consumption parameters. A server is treated as an M/G/1 queueing system with mixed classes of tasks arriving in a Poisson stream (see Figure 1).

Assume that the tasks of each class of applications arrive according to a Poisson process. The arrival rate of the i th type of applications is $\tilde{\lambda}_i$, where $1 \leq i \leq m$. The total task arrival rate is $\lambda = \tilde{\lambda}_1 + \tilde{\lambda}_2 + \dots + \tilde{\lambda}_m$. A load distribution mechanism splits the stream of the i th class of applications into n substreams with rates $\lambda_{i,1}, \lambda_{i,2}, \dots, \lambda_{i,n}$, such that the substream of tasks with rate $\lambda_{i,j}$ is assigned to server j . The total task arrival rate to server j is $\lambda_j = \lambda_{1,j} + \lambda_{2,j} + \dots + \lambda_{m,j}$, where $1 \leq j \leq n$.

For the i th type of applications, the execution requirements of the tasks are independent and identically distributed (i.i.d.) random variables r_i with mean \bar{r}_i and second moment \bar{r}_i^2 . The execution speed for the i th type of applications on server j is $s_{i,j}$. Hence, the execution times of the tasks of the i th type of applications on server j are i.i.d. random variables $x_{i,j} = r_i/s_{i,j}$ with mean $\bar{x}_{i,j} = \bar{r}_i/s_{i,j}$ and second moment $\bar{x}_{i,j}^2 = \bar{r}_i^2/s_{i,j}^2$, where $1 \leq i \leq m$ and $1 \leq j \leq n$.

The execution time of a task on server j is a random variable x_j with mean

$$\bar{x}_j = \frac{\lambda_{1,j}}{\lambda_j} \bar{x}_{1,j} + \frac{\lambda_{2,j}}{\lambda_j} \bar{x}_{2,j} + \dots + \frac{\lambda_{m,j}}{\lambda_j} \bar{x}_{m,j}.$$

The utilization of server j is

$$\rho_j = \lambda_j \bar{x}_j = \lambda_{1,j} \bar{x}_{1,j} + \lambda_{2,j} \bar{x}_{2,j} + \dots + \lambda_{m,j} \bar{x}_{m,j}.$$

The second moment of x_j is

$$\bar{x}_j^2 = \frac{\lambda_{1,j}}{\lambda_j} \bar{x}_{1,j}^2 + \frac{\lambda_{2,j}}{\lambda_j} \bar{x}_{2,j}^2 + \dots + \frac{\lambda_{m,j}}{\lambda_j} \bar{x}_{m,j}^2.$$

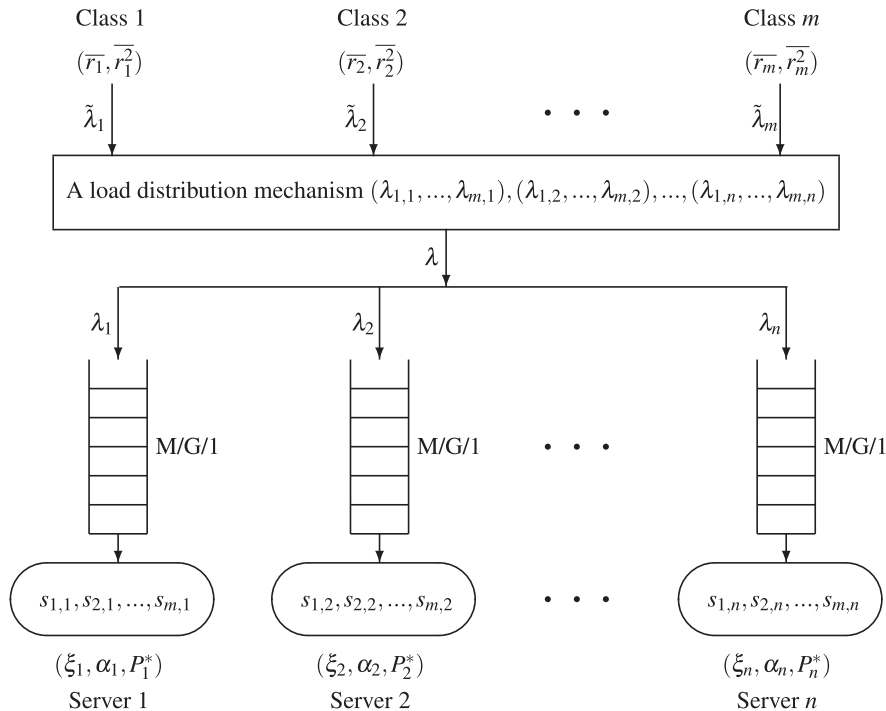


FIGURE 1 Load distribution on heterogeneous servers with variable speeds

The average waiting time of a task on server j is (See p. 190 in the work of Kleinrock⁴⁵)

$$W_j = \frac{\lambda_j \overline{x_j^2}}{2(1 - \rho_j)} = \frac{\sigma_j}{2(1 - \rho_j)},$$

where

$$\sigma_j = \lambda_j \overline{x_j^2} = \lambda_{1,j} \overline{x_{1,j}^2} + \lambda_{2,j} \overline{x_{2,j}^2} + \cdots + \lambda_{m,j} \overline{x_{m,j}^2}.$$

The average response time of the tasks of the i th type of applications on server j is

$$T_{i,j} = \overline{x_{i,j}} + W_j = \overline{x_{i,j}} + \frac{\sigma_j}{2(1 - \rho_j)},$$

which can be rewritten as

$$T_{i,j} = \overline{x_{i,j}} + \frac{\lambda_{1,j} \overline{x_{1,j}^2} + \lambda_{2,j} \overline{x_{2,j}^2} + \cdots + \lambda_{m,j} \overline{x_{m,j}^2}}{2(1 - \lambda_{1,j} \overline{x_{1,j}} - \lambda_{2,j} \overline{x_{2,j}} - \cdots - \lambda_{m,j} \overline{x_{m,j}})},$$

and

$$T_{i,j} = \frac{\overline{r_i}}{s_{i,j}} + \frac{\lambda_{1,j} \overline{x_{1,j}^2} + \lambda_{2,j} \overline{x_{2,j}^2} + \cdots + \lambda_{m,j} \overline{x_{m,j}^2}}{2(1 - \lambda_{1,j} \overline{x_{1,j}} - \lambda_{2,j} \overline{x_{2,j}} - \cdots - \lambda_{m,j} \overline{x_{m,j}})},$$

where $1 \leq i \leq m$ and $1 \leq j \leq n$.

The average response time of all tasks on server j is

$$T_j = \sum_{i=1}^m \frac{\lambda_{i,j}}{\lambda_j} T_{i,j} = \frac{1}{\lambda_j} \sum_{i=1}^m \frac{\lambda_{i,j} \overline{r_i}}{s_{i,j}} + \frac{\sigma_j}{2(1 - \rho_j)} = \frac{\rho_j}{\lambda_j} + \frac{\sigma_j}{2(1 - \rho_j)},$$

which is actually

$$T_j = \overline{x_j} + W_j,$$

where

$$\overline{x_j} = \frac{\rho_j}{\lambda_j} = \frac{1}{\lambda_j} \left(\frac{\lambda_{1,j} \overline{r_1}}{s_{1,j}} + \frac{\lambda_{2,j} \overline{r_2}}{s_{2,j}} + \cdots + \frac{\lambda_{m,j} \overline{r_m}}{s_{m,j}} \right),$$

and

$$\rho_j = \lambda_{1,j} \frac{\overline{r_1}}{s_{1,j}} + \lambda_{2,j} \frac{\overline{r_2}}{s_{2,j}} + \cdots + \lambda_{m,j} \frac{\overline{r_m}}{s_{m,j}},$$

and

$$\sigma_j = \lambda_{1,j} \frac{\overline{r_1^2}}{s_{1,j}^2} + \lambda_{2,j} \frac{\overline{r_2^2}}{s_{2,j}^2} + \cdots + \lambda_{m,j} \frac{\overline{r_m^2}}{s_{m,j}^2},$$

where $1 \leq j \leq n$. The average response time of all tasks on the n servers is

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

which is the main performance goal to be optimized.

Assume that server j has a base power consumption P_j^* and consumes no dynamic power when it is idle. The dynamic power consumption of server j is $\xi_j s^{\alpha_j}$ when its speed is s . The average power consumption of server j is

$$P_j = \sum_{i=1}^m \lambda_{i,j} \overline{x_{i,j}} \xi_j s_{i,j}^{\alpha_j} + P_j^* = \xi_j \sum_{i=1}^m \lambda_{i,j} \overline{r_i} s_{i,j}^{\alpha_j - 1} + P_j^*,$$

where $1 \leq j \leq n$. The total power consumption of the n servers is

$$P = P_1 + P_2 + \cdots + P_n,$$

which is another performance goal to be optimized.

4 | POWER CONSTRAINED PERFORMANCE OPTIMIZATION

Our power constrained performance optimization problem can be defined as follows. Given m classes of applications with task arrival rates $\bar{\lambda}_1, \bar{\lambda}_2, \dots, \bar{\lambda}_m$, expected task execution requirements $\bar{r}_1, \bar{r}_2, \dots, \bar{r}_m$, the second moments of task execution requirements $\bar{r}_1^2, \bar{r}_2^2, \dots, \bar{r}_m^2$, and n heterogeneous servers with coefficients $\xi_1, \xi_2, \dots, \xi_n$ and exponents $\alpha_1, \alpha_2, \dots, \alpha_n$ for dynamic power consumption, and base power $P_1^*, P_2^*, \dots, P_n^*$ for static power consumption, and certain power supply \bar{P} , our problem is to find load distribution $\lambda_{i,j}$ and server speeds $s_{i,j}$, for all $1 \leq i \leq m$ and $1 \leq j \leq n$, such that T is minimized and that P does not exceed \bar{P} . The optimization problem contains $3(m+n)+1$ input parameters and $2mn$ output parameters. It needs to determine an optimal load distribution for multiple classes of applications over heterogeneous servers and an optimal application-dependent server speed setting for all the servers.

The aforementioned optimization problem is a multivariable (ie, $2mn$ variables) optimization problem with multiple (ie, $m+1$) constraints, ie, $L_i = \lambda_{i,1} + \lambda_{i,2} + \dots + \lambda_{i,n} = \bar{\lambda}_i$, for all $1 \leq i \leq m$, and $P = \bar{P}$. Notice that T , P , and L_i are all treated as functions of the $\lambda_{i,j}$'s and the $s_{i,j}$'s. The optimization problem can be solved by using the method of Lagrange multiplier, namely,

$$\nabla T = \phi_1 \nabla L_1 + \phi_2 \nabla L_2 + \dots + \phi_m \nabla L_m + \psi \nabla P,$$

that is,

$$\frac{\partial T}{\partial \lambda_{i,j}} = \phi_1 \frac{\partial L_1}{\partial \lambda_{i,j}} + \phi_2 \frac{\partial L_2}{\partial \lambda_{i,j}} + \dots + \phi_m \frac{\partial L_m}{\partial \lambda_{i,j}} + \psi \frac{\partial P}{\partial \lambda_{i,j}},$$

and

$$\frac{\partial T}{\partial s_{i,j}} = \phi_1 \frac{\partial L_1}{\partial s_{i,j}} + \phi_2 \frac{\partial L_2}{\partial s_{i,j}} + \dots + \phi_m \frac{\partial L_m}{\partial s_{i,j}} + \psi \frac{\partial P}{\partial s_{i,j}},$$

for all $1 \leq i \leq m$ and $1 \leq j \leq n$, where $\phi_1, \phi_2, \dots, \phi_m$ and ψ are $m+1$ Lagrange multipliers.

In the following, we calculate all the partial derivatives and transform our optimization problem into a system of nonlinear equations. Notice that

$$T = \frac{1}{\lambda} \sum_{j=1}^n \lambda_j T_j = \frac{1}{\lambda} \sum_{j=1}^n \left(\rho_j + \frac{\lambda_j \sigma_j}{2(1-\rho_j)} \right).$$

Since

$$\frac{\partial \lambda}{\partial \lambda_{i,j}} = \frac{\partial \lambda_j}{\partial \lambda_{i,j}} = 1,$$

and

$$\frac{\partial \sigma_j}{\partial \lambda_{i,j}} = \frac{\bar{r}_i^2}{s_{i,j}^2},$$

and

$$\frac{\partial \rho_j}{\partial \lambda_{i,j}} = \frac{\bar{r}_i}{s_{i,j}},$$

we have

$$\frac{\partial T}{\partial \lambda_{i,j}} = -\frac{1}{\lambda^2} \sum_{j=1}^n \left(\rho_j + \frac{\lambda_j \sigma_j}{2(1-\rho_j)} \right) + \frac{1}{\lambda} \left(\frac{\bar{r}_i}{s_{i,j}} + \frac{1}{2} \left(\frac{\sigma_j}{1-\rho_j} + \frac{\lambda_j}{1-\rho_j} \cdot \frac{\bar{r}_i^2}{s_{i,j}^2} + \frac{\lambda_j \sigma_j}{(1-\rho_j)^2} \cdot \frac{\bar{r}_i}{s_{i,j}} \right) \right).$$

Furthermore, since $\partial L_i / \partial \lambda_{i,j} = 1$ and $\partial L_{i'} / \partial \lambda_{i,j} = 0$, for all $i' \neq i$, and

$$\frac{\partial P}{\partial \lambda_{i,j}} = \frac{\partial P_j}{\partial \lambda_{i,j}} = \xi_j \bar{r}_i s_{i,j}^{\alpha_j - 1},$$

we have

$$-\frac{1}{\lambda^2} \sum_{j=1}^n \left(\rho_j + \frac{\lambda_j \sigma_j}{2(1-\rho_j)} \right) + \frac{1}{\lambda} \left(\frac{\bar{r}_i}{s_{i,j}} + \frac{1}{2} \left(\frac{\sigma_j}{1-\rho_j} + \frac{\lambda_j}{1-\rho_j} \cdot \frac{\bar{r}_i^2}{s_{i,j}^2} + \frac{\lambda_j \sigma_j}{(1-\rho_j)^2} \cdot \frac{\bar{r}_i}{s_{i,j}} \right) \right) = \phi_i + \psi \xi_j \bar{r}_i s_{i,j}^{\alpha_j - 1},$$

for all $1 \leq i \leq m$ and $1 \leq j \leq n$. Notice that $\phi_i > 0$ because $\partial T / \partial \lambda_{i,j} > 0$ (ie, T is an increasing function of $\lambda_{i,j}$) and $\psi < 0$ (see as follows). The last equation can be rewritten as

$$G_{i,j} = \frac{1}{\lambda^2} \sum_{j=1}^n \left(\rho_j + \frac{\lambda_j \sigma_j}{2(1-\rho_j)} \right) - \frac{1}{\lambda} \left(\frac{\bar{r}_i}{s_{i,j}} + \frac{1}{2} \left(\frac{\sigma_j}{1-\rho_j} + \frac{\lambda_j}{1-\rho_j} \cdot \frac{\bar{r}_i^2}{s_{i,j}^2} + \frac{\lambda_j \sigma_j}{(1-\rho_j)^2} \cdot \frac{\bar{r}_i}{s_{i,j}} \right) \right) + \phi_i + \psi \xi_j \bar{r}_i s_{i,j}^{\alpha_j - 1} = 0,$$

for all $1 \leq i \leq m$ and $1 \leq j \leq n$.

Since

$$\frac{\partial \sigma_j}{\partial s_{i,j}} = -\frac{2\lambda_{i,j}\bar{r}_i^2}{s_{i,j}^3},$$

and

$$\frac{\partial \rho_j}{\partial s_{i,j}} = -\frac{\lambda_{i,j}\bar{r}_i}{s_{i,j}^2},$$

we have

$$\frac{\partial T}{\partial s_{i,j}} = \frac{1}{\lambda} \left(-\frac{\lambda_{i,j}\bar{r}_i}{s_{i,j}^2} + \frac{\lambda_j}{2} \left(\frac{1}{1-\rho_j} \left(-\frac{2\lambda_{i,j}\bar{r}_i^2}{s_{i,j}^3} \right) + \frac{\sigma_j}{(1-\rho_j)^2} \left(-\frac{\lambda_{i,j}\bar{r}_i}{s_{i,j}^2} \right) \right) \right).$$

Furthermore, since $\partial L_{i'}/\partial s_{i,j} = 0$, for all $1 \leq i' \leq m$, and

$$\frac{\partial P}{\partial s_{i,j}} = \frac{\partial P_j}{\partial s_{i,j}} = \xi_j \lambda_{i,j} \bar{r}_i (\alpha_j - 1) s_{i,j}^{\alpha_j - 2},$$

we have

$$\frac{1}{\lambda} \left(-\frac{\lambda_{i,j}\bar{r}_i}{s_{i,j}^2} + \frac{\lambda_j}{2} \left(\frac{1}{1-\rho_j} \left(-\frac{2\lambda_{i,j}\bar{r}_i^2}{s_{i,j}^3} \right) + \frac{\sigma_j}{(1-\rho_j)^2} \left(-\frac{\lambda_{i,j}\bar{r}_i}{s_{i,j}^2} \right) \right) \right) = \psi \xi_j \lambda_{i,j} \bar{r}_i (\alpha_j - 1) s_{i,j}^{\alpha_j - 2},$$

for all $1 \leq i \leq m$ and $1 \leq j \leq n$. Notice that $\psi < 0$, since $\partial T/\partial s_{i,j} < 0$, ie, T is a decreasing function of $s_{i,j}$. The last equation can be rewritten as

$$-\frac{1}{\lambda} \left(\frac{\lambda_{i,j}\bar{r}_i}{s_{i,j}^2} + \frac{\lambda_j}{2} \left(\frac{1}{1-\rho_j} \cdot \frac{2\lambda_{i,j}\bar{r}_i^2}{s_{i,j}^3} + \frac{\sigma_j}{(1-\rho_j)^2} \cdot \frac{\lambda_{i,j}\bar{r}_i}{s_{i,j}^2} \right) \right) = \psi \xi_j \lambda_{i,j} \bar{r}_i (\alpha_j - 1) s_{i,j}^{\alpha_j - 2},$$

or

$$-\frac{1}{\lambda} \left(1 + \frac{\lambda_j}{2} \left(\frac{\bar{r}_i^2}{\bar{r}_i} \cdot \frac{1}{1-\rho_j} \cdot \frac{2}{s_{i,j}} + \frac{\sigma_j}{(1-\rho_j)^2} \right) \right) = \psi \xi_j (\alpha_j - 1) s_{i,j}^{\alpha_j},$$

or

$$H_{i,j} = \frac{1}{\lambda} \left(1 + \frac{\lambda_j}{2} \left(\frac{\bar{r}_i^2}{\bar{r}_i} \cdot \frac{1}{1-\rho_j} \cdot \frac{2}{s_{i,j}} + \frac{\sigma_j}{(1-\rho_j)^2} \right) \right) + \psi \xi_j (\alpha_j - 1) s_{i,j}^{\alpha_j} = 0,$$

for all $1 \leq i \leq m$ and $1 \leq j \leq n$.

The aforementioned equations for $G_{i,j}$ and $H_{i,j}$, together with

$$J_i = L_i - \tilde{\lambda}_i = \lambda_{i,1} + \lambda_{i,2} + \dots + \lambda_{i,n} - \tilde{\lambda}_i = 0,$$

for all $1 \leq i \leq m$, and

$$K = P - \tilde{P} = \sum_{j=1}^n \left(\xi_j \sum_{i=1}^m \lambda_{i,j} \bar{r}_i s_{i,j}^{\alpha_j - 1} + P_j^* \right) - \tilde{P} = 0,$$

constitute a system of $2mn + m + 1$ nonlinear equations with $2mn + m + 1$ unknowns, ie, the $\lambda_{i,j}$'s, the $s_{i,j}$'s, the ϕ_i 's, and ψ .

5 | A NUMERICAL ALGORITHM

The discussion in the last section gives rise to a system of nonlinear equations, ie,

$$\begin{aligned} F_1(y_1, y_2, \dots, y_N) &= 0, \\ F_2(y_1, y_2, \dots, y_N) &= 0, \\ &\vdots \\ F_N(y_1, y_2, \dots, y_N) &= 0, \end{aligned}$$

where $N = 2mn + m + 1$, $y_{(i-1)n+j} = \lambda_{i,j}$, $y_{mn+(i-1)n+j} = s_{i,j}$, $y_{2mn+i} = \phi_i$, $y_N = \psi$, $F_{(i-1)n+j} = G_{i,j}$, $F_{mn+(i-1)n+j} = H_{i,j}$, $F_{2mn+i} = J_i$, and $F_N = K$, for all $1 \leq i \leq m$ and $1 \leq j \leq n$.

By using vector notation to represent the N variables, ie, the $\lambda_{i,j}$'s, the $s_{i,j}$'s, the ϕ_i 's, and ψ , we write

$$\mathbf{y} = (y_1, y_2, \dots, y_N) = (\lambda_{1,1}, \dots, \lambda_{m,n}, s_{1,1}, \dots, s_{m,n}, \phi_1, \dots, \phi_m, \psi),$$

and $F_k(y_1, y_2, \dots, y_N) = F_k(\mathbf{y})$, where $F_k : \mathbb{R}^N \rightarrow \mathbb{R}$ maps N -dimensional space \mathbb{R}^N into the real line \mathbb{R} . By defining a function $\mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$, which maps \mathbb{R}^N into \mathbb{R}^N ,

$$\mathbf{F}(\mathbf{y}) = (F_1(y_1, y_2, \dots, y_N), F_2(y_1, y_2, \dots, y_N), \dots, F_N(y_1, y_2, \dots, y_N)),$$

namely,

$$\mathbf{F}(\mathbf{y}) = (F_1(\mathbf{y}), F_2(\mathbf{y}), \dots, F_N(\mathbf{y})),$$

then our system of nonlinear equations is

$$\mathbf{F}(\mathbf{y}) = \mathbf{0},$$

where $\mathbf{0} = (0, 0, \dots, 0)$.

The aforementioned system of nonlinear equations can be solved by using Newton's method. To this end, we need the Jacobian matrix $\mathbf{J}(\mathbf{y})$ defined as

$$\mathbf{J}(\mathbf{y}) = \begin{bmatrix} \frac{\partial F_1(\mathbf{y})}{\partial y_1} & \frac{\partial F_1(\mathbf{y})}{\partial y_2} & \dots & \frac{\partial F_1(\mathbf{y})}{\partial y_N} \\ \frac{\partial F_2(\mathbf{y})}{\partial y_1} & \frac{\partial F_2(\mathbf{y})}{\partial y_2} & \dots & \frac{\partial F_2(\mathbf{y})}{\partial y_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_N(\mathbf{y})}{\partial y_1} & \frac{\partial F_N(\mathbf{y})}{\partial y_2} & \dots & \frac{\partial F_N(\mathbf{y})}{\partial y_N} \end{bmatrix}.$$

The various components of the aforementioned matrix are calculated as follows. As seen from the aforementioned discussion, our unknowns and equations are divided into 4 groups. For clarity, we only show $\partial F_i(\mathbf{y})/\partial y_j$ if it is not zero. As a default, a component is zero if it is not mentioned in the following computation.

First, we consider F_k , for $1 \leq k \leq mn$. Let $k = (i-1)n + j$ and $F_k = G_{i,j}$.

- We have

$$\begin{aligned} \frac{\partial G_{i,j}}{\partial \lambda_{i',j}} &= -\frac{2}{\lambda^3} \sum_{j=1}^n \left(\rho_j + \frac{\lambda_j \sigma_j}{2(1-\rho_j)} \right) \\ &+ \frac{1}{\lambda^2} \left(\frac{\bar{r}_{i'}}{s_{i',j}} + \frac{1}{2} \left(\frac{\sigma_j}{1-\rho_j} + \frac{\lambda_j}{1-\rho_j} \cdot \frac{\bar{r}_{i'}^2}{s_{i',j}^2} + \frac{\lambda_j \sigma_j}{(1-\rho_j)^2} \cdot \frac{\bar{r}_{i'}}{s_{i',j}} \right) \right) \\ &+ \frac{1}{\lambda^2} \left(\frac{\bar{r}_i}{s_{i,j}} + \frac{1}{2} \left(\frac{\sigma_j}{1-\rho_j} + \frac{\lambda_j}{1-\rho_j} \cdot \frac{\bar{r}_i^2}{s_{i,j}^2} + \frac{\lambda_j \sigma_j}{(1-\rho_j)^2} \cdot \frac{\bar{r}_i}{s_{i,j}} \right) \right) \\ &- \frac{1}{2\lambda} \left(\frac{1}{1-\rho_j} \cdot \frac{\bar{r}_{i'}^2}{s_{i',j}^2} + \frac{\sigma_j}{(1-\rho_j)^2} \cdot \frac{\bar{r}_{i'}}{s_{i',j}} + \frac{\bar{r}_i^2}{s_{i,j}^2} \left(\frac{1}{1-\rho_j} + \frac{\lambda_j}{(1-\rho_j)^2} \cdot \frac{\bar{r}_{i'}}{s_{i',j}} \right) \right) \\ &+ \frac{\bar{r}_i}{s_{i,j}} \left(\frac{\sigma_j}{(1-\rho_j)^2} + \frac{\lambda_j}{(1-\rho_j)^2} \cdot \frac{\bar{r}_{i'}^2}{s_{i',j}^2} + \frac{2\lambda_j \sigma_j}{(1-\rho_j)^3} \cdot \frac{\bar{r}_{i'}}{s_{i',j}} \right), \end{aligned}$$

for all $1 \leq i' \leq m$, and

$$\begin{aligned} \frac{\partial G_{i,j}}{\partial \lambda_{i',j'}} &= -\frac{2}{\lambda^3} \sum_{j=1}^n \left(\rho_j + \frac{\lambda_j \sigma_j}{2(1-\rho_j)} \right) \\ &\quad + \frac{1}{\lambda^2} \left(\frac{\bar{r}_{i'}}{s_{i',j'}} + \frac{1}{2} \left(\frac{\sigma_{j'}}{1-\rho_{j'}} + \frac{\lambda_{j'}}{1-\rho_{j'}} \cdot \frac{\bar{r}_{i'}^2}{s_{i',j'}^2} + \frac{\lambda_{j'} \sigma_{j'}}{(1-\rho_{j'})^2} \cdot \frac{\bar{r}_{i'}}{s_{i',j'}} \right) \right) \\ &\quad + \frac{1}{\lambda^2} \left(\frac{\bar{r}_i}{s_{i,j}} + \frac{1}{2} \left(\frac{\sigma_j}{1-\rho_j} + \frac{\lambda_j}{1-\rho_j} \cdot \frac{\bar{r}_i^2}{s_{i,j}^2} + \frac{\lambda_j \sigma_j}{(1-\rho_j)^2} \cdot \frac{\bar{r}_i}{s_{i,j}} \right) \right), \end{aligned}$$

for all $1 \leq i' \leq m$ and $j' \neq j$.

- We have

$$\begin{aligned} \frac{\partial G_{i,j}}{\partial s_{i,j}} &= \frac{1}{\lambda^2} \left(-\frac{\lambda_{i,j} \bar{r}_i}{s_{i,j}^2} + \frac{\lambda_j}{2} \left(\frac{1}{1-\rho_j} \left(-\frac{2\lambda_{i,j} \bar{r}_i^2}{s_{i,j}^3} \right) + \frac{\sigma_j}{(1-\rho_j)^2} \left(-\frac{\lambda_{i,j} \bar{r}_i}{s_{i,j}^2} \right) \right) \right) \\ &\quad - \frac{1}{\lambda} \left(-\frac{\bar{r}_i}{s_{i,j}^2} + \frac{1}{2} \left(\frac{1}{1-\rho_j} \left(-\frac{2\lambda_{i,j} \bar{r}_i^2}{s_{i,j}^3} \right) + \frac{\sigma_j}{(1-\rho_j)^2} \left(-\frac{\lambda_{i,j} \bar{r}_i}{s_{i,j}^2} \right) \right) \right) \\ &\quad + \lambda_j \bar{r}_i^2 \left(\frac{1}{(1-\rho_j)^2} \cdot \frac{1}{s_{i,j}^2} \left(-\frac{\lambda_{i,j} \bar{r}_i}{s_{i,j}^2} \right) + \frac{1}{1-\rho_j} \left(-\frac{2}{s_{i,j}^3} \right) \right) \\ &\quad + \lambda_j \bar{r}_i \left(\frac{1}{(1-\rho_j)^2} \cdot \frac{1}{s_{i,j}} \left(-\frac{2\lambda_{i,j} \bar{r}_i^2}{s_{i,j}^3} \right) + \frac{2\sigma_j}{(1-\rho_j)^3} \cdot \frac{1}{s_{i,j}} \left(-\frac{\lambda_{i,j} \bar{r}_i}{s_{i,j}^2} \right) + \frac{\sigma_j}{(1-\rho_j)^2} \left(-\frac{1}{s_{i,j}^2} \right) \right) \Bigg) \\ &\quad + \psi \xi_j \bar{r}_i (\alpha_j - 1) s_{i,j}^{\alpha_j - 2}, \end{aligned}$$

and

$$\begin{aligned} \frac{\partial G_{i,j}}{\partial s_{i',j}} &= \frac{1}{\lambda^2} \left(-\frac{\lambda_{i',j} \bar{r}_{i'}}{s_{i',j}^2} + \frac{\lambda_j}{2} \left(\frac{1}{1-\rho_j} \left(-\frac{2\lambda_{i',j} \bar{r}_{i'}^2}{s_{i',j}^3} \right) + \frac{\sigma_j}{(1-\rho_j)^2} \left(-\frac{\lambda_{i',j} \bar{r}_{i'}}{s_{i',j}^2} \right) \right) \right) \\ &\quad - \frac{1}{2\lambda} \left(\frac{1}{1-\rho_j} \left(-\frac{2\lambda_{i',j} \bar{r}_{i'}^2}{s_{i',j}^3} \right) + \frac{\sigma_j}{(1-\rho_j)^2} \left(-\frac{\lambda_{i',j} \bar{r}_{i'}}{s_{i',j}^2} \right) \right) \\ &\quad + \frac{\lambda_j \bar{r}_i^2}{s_{i,j}^2} \cdot \frac{1}{(1-\rho_j)^2} \left(-\frac{\lambda_{i',j} \bar{r}_{i'}}{s_{i',j}^2} \right) \\ &\quad + \frac{\lambda_j \bar{r}_i}{s_{i,j}} \left(\frac{1}{(1-\rho_j)^2} \left(-\frac{2\lambda_{i',j} \bar{r}_{i'}^2}{s_{i',j}^3} \right) + \frac{2\sigma_j}{(1-\rho_j)^3} \left(-\frac{\lambda_{i',j} \bar{r}_{i'}}{s_{i',j}^2} \right) \right) \Bigg), \end{aligned}$$

for all $i' \neq i$, and

$$\frac{\partial G_{i,j}}{\partial s_{i',j'}} = \frac{1}{\lambda^2} \left(-\frac{\lambda_{i',j'} \bar{r}_{i'}}{s_{i',j'}^2} + \frac{\lambda_{j'}}{2} \left(\frac{1}{1-\rho_{j'}} \left(-\frac{2\lambda_{i',j'} \bar{r}_{i'}^2}{s_{i',j'}^3} \right) + \frac{\sigma_{j'}}{(1-\rho_{j'})^2} \left(-\frac{\lambda_{i',j'} \bar{r}_{i'}}{s_{i',j'}^2} \right) \right) \right),$$

for all $1 \leq i' \leq m$ and $j' \neq j$.

- We have $\partial G_{i,j} / \phi_i = 1$.
- We have $\partial G_{i,j} / \psi = \xi_j \bar{r}_i s_{i,j}^{\alpha_j - 1}$.

Next, we consider F_k , for $mn + 1 \leq k \leq 2mn$. Let $k = mn + (i-1)n + j$ and $F_k = H_{i,j}$.

- We have

$$\begin{aligned} \frac{\partial H_{i,j}}{\partial \lambda_{i',j}} = & -\frac{1}{\lambda^2} \left(1 + \frac{\lambda_j}{2} \left(\frac{\bar{r}_i^2}{\bar{r}_i} \cdot \frac{1}{1-\rho_j} \cdot \frac{2}{s_{i,j}} + \frac{\sigma_j}{(1-\rho_j)^2} \right) \right) \\ & + \frac{1}{\lambda} \left(\frac{1}{2} \left(\frac{\bar{r}_i^2}{\bar{r}_i} \cdot \frac{1}{1-\rho_j} \cdot \frac{2}{s_{i,j}} + \frac{\sigma_j}{(1-\rho_j)^2} \right) \right) \\ & + \frac{\lambda_j}{2} \left(\frac{\bar{r}_i^2}{\bar{r}_i} \cdot \frac{1}{(1-\rho_j)^2} \cdot \frac{2}{s_{i,j}} \cdot \frac{\bar{r}_{i'}}{s_{i',j}} + \frac{1}{(1-\rho_j)^2} \cdot \frac{\bar{r}_{i'}^2}{s_{i',j}^2} + \frac{2\sigma_j}{(1-\rho_j)^3} \cdot \frac{\bar{r}_{i'}}{s_{i',j}} \right), \end{aligned}$$

for all $1 \leq i' \leq m$, and

$$\frac{\partial H_{i,j}}{\partial \lambda_{i',j'}} = -\frac{1}{\lambda^2} \left(1 + \frac{\lambda_j}{2} \left(\frac{\bar{r}_i^2}{\bar{r}_i} \cdot \frac{1}{1-\rho_j} \cdot \frac{2}{s_{i,j}} + \frac{\sigma_j}{(1-\rho_j)^2} \right) \right),$$

for all $1 \leq i' \leq m$ and $j' \neq j$.

- We have

$$\begin{aligned} \frac{\partial H_{i,j}}{\partial s_{i,j}} = & \frac{1}{\lambda} \cdot \frac{\lambda_j}{2} \left(\frac{\bar{r}_i^2}{\bar{r}_i} \left(\frac{1}{(1-\rho_j)^2} \cdot \frac{2}{s_{i,j}} \left(-\frac{\lambda_{i,j} \bar{r}_i}{s_{i,j}^2} \right) + \frac{1}{1-\rho_j} \left(-\frac{2}{s_{i,j}^2} \right) \right) \right) \\ & + \frac{1}{(1-\rho_j)^2} \left(-\frac{2\lambda_{i,j} \bar{r}_i^2}{s_{i,j}^3} \right) + \frac{2\sigma_j}{(1-\rho_j)^3} \left(-\frac{\lambda_{i,j} \bar{r}_i}{s_{i,j}^2} \right) \\ & + \psi \xi_j \alpha_j (\alpha_j - 1) s_{i,j}^{\alpha_j - 1}, \end{aligned}$$

and

$$\begin{aligned} \frac{\partial H_{i,j}}{\partial s_{i',j}} = & \frac{1}{\lambda} \cdot \frac{\lambda_j}{2} \left(\frac{\bar{r}_i^2}{\bar{r}_i} \cdot \frac{1}{(1-\rho_j)^2} \cdot \frac{2}{s_{i,j}} \left(-\frac{\lambda_{i',j} \bar{r}_{i'}}{s_{i',j}^2} \right) \right) \\ & + \frac{1}{(1-\rho_j)^2} \left(-\frac{2\lambda_{i',j} \bar{r}_{i'}^2}{s_{i',j}^3} \right) + \frac{2\sigma_j}{(1-\rho_j)^3} \left(-\frac{\lambda_{i',j} \bar{r}_{i'}}{s_{i',j}^2} \right), \end{aligned}$$

for all $i' \neq i$.

- We have $\partial H_{i,j} / \partial \psi = \xi_j (\alpha_j - 1) s_{i,j}^{\alpha_j}$.

Third, we consider F_k , for $2mn + 1 \leq k \leq 2mn + m$. Let $k = 2mn + i$ and $F_k = J_i$.

- We have $\partial J_i / \partial \lambda_{i,j} = 1$, for all $1 \leq j \leq n$.

Finally, we consider $F_N = K$.

- We have $\partial K / \partial \lambda_{i,j} = \xi_j \bar{r}_i s_{i,j}^{\alpha_j - 1}$, for all $1 \leq i \leq m$ and $1 \leq j \leq n$.
- We have $\partial K / \partial s_{i,j} = \xi_j \lambda_{i,j} \bar{r}_i (\alpha_j - 1) s_{i,j}^{\alpha_j - 2}$, for all $1 \leq i \leq m$ and $1 \leq j \leq n$.

Our numerical algorithm for finding an optimal load distribution $\lambda_{i,j}$, an optimal server speed setting $s_{i,j}$, and the Lagrange multipliers $\phi_1, \phi_2, \dots, \phi_m, \psi$, ie, the vector $\mathbf{y} = (y_1, y_2, \dots, y_N)$ that satisfies the system of nonlinear equations $\mathbf{F}(\mathbf{y}) = \mathbf{0}$, is given in Algorithm 1. This is essentially the standard Newton's iterative method (See p. 451 in the work of Burden et al⁴⁶). Our initial approximation of \mathbf{y} is $\lambda_{i,j} = \tilde{\lambda}_i / n$, $s_{i,j} = s$, $\phi_i = 1$, and $\psi = -1$, for all $1 \leq i \leq m$ and $1 \leq j \leq n$ (line(1)), where s is a constant speed of the servers, which satisfies

$$\sum_{j=1}^n \left(\xi_j \sum_{i=1}^m \lambda_{i,j} \bar{r}_i s_{i,j}^{\alpha_j - 1} + P_j^* \right) = \sum_{j=1}^n \left(\frac{\xi_j}{n} \sum_{i=1}^m \tilde{\lambda}_i \bar{r}_i s_{i,j}^{\alpha_j - 1} + P_j^* \right) = \tilde{P},$$

Algorithm 1: An algorithm for finding an optimal load distribution and server speed setting

Input: $\tilde{\lambda}_1, \tilde{\lambda}_2, \dots, \tilde{\lambda}_m, \bar{r}_1, \bar{r}_2, \dots, \bar{r}_m, \bar{r}_1^2, \bar{r}_2^2, \dots, \bar{r}_m^2, \xi_1, \xi_2, \dots, \xi_n, \alpha_1, \alpha_2, \dots, \alpha_n, P_1^*, P_2^*, \dots, P_n^*$, and \tilde{P} .

Output: $\lambda_{1,1}, \dots, \lambda_{m,n}, s_{1,1}, \dots, s_{m,n}, \phi_1, \dots, \phi_m, \psi$, i.e., $\mathbf{y} = (y_1, y_2, \dots, y_N)$, which satisfies $\mathbf{F}(\mathbf{y}) = \mathbf{0}$.

$\mathbf{y} \leftarrow (\tilde{\lambda}_1/n, \dots, \tilde{\lambda}_m/n, s, \dots, s, 1, \dots, 1, -1);$ (1)
repeat (2)
 Calculate $\mathbf{J}(\mathbf{y})$, where $\mathbf{J}(\mathbf{y})_{i,j} = \partial F_i(\mathbf{y})/\partial y_j$ for $1 \leq i, j \leq N$; (3)
 Calculate $\mathbf{F}(\mathbf{y}) = (F_1(\mathbf{y}), F_2(\mathbf{y}), \dots, F_N(\mathbf{y}))$; (4)
 Solve the system of linear equations $\mathbf{J}(\mathbf{y})\mathbf{z} = -\mathbf{F}(\mathbf{y})$; (5)
 $\mathbf{y} \leftarrow \mathbf{y} + \mathbf{z}$; (6)
until $\|\mathbf{z}\| \leq \epsilon$. (7)

that is,

$$\frac{1}{n} \left(\sum_{i=1}^m \tilde{\lambda}_i \bar{r}_i \right) \left(\sum_{j=1}^n \xi_j s^{\alpha_j - 1} \right) = \tilde{P} - \sum_{j=1}^n P_j^*.$$

The value of \mathbf{y} is then repeatedly modified as $\mathbf{y} + \mathbf{z}$ (line (6)), where \mathbf{z} is the solution to the system of linear equations $\mathbf{J}(\mathbf{y})\mathbf{z} = -\mathbf{F}(\mathbf{y})$ (line (5)). Such modification is repeated until $\|\mathbf{z}\| \leq \epsilon$ (line (7)), where

$$\|\mathbf{z}\| = \sqrt{z_1^2 + z_2^2 + \dots + z_N^2},$$

and ϵ is a sufficiently small constant, say, 10^{-10} . The system of linear equations in line (5) can be solved by using the classic Gaussian elimination with backward substitution algorithm (See pp. 268-269 in the work of Burden et al⁴⁶).

The value of s that satisfies

$$\sum_{j=1}^n \xi_j s^{\alpha_j - 1} = \left(\tilde{P} - \sum_{j=1}^n P_j^* \right) / \left(\frac{1}{n} \sum_{i=1}^m \tilde{\lambda}_i \bar{r}_i \right)$$

can be found by using the standard bisection method (See p. 22 in the work of Burden et al⁴⁶), ie, searching s in an interval $[s', s'']$. Let $\alpha' = \min\{\alpha_1, \alpha_2, \dots, \alpha_n\}$ and $\alpha'' = \max\{\alpha_1, \alpha_2, \dots, \alpha_n\}$. Since

$$\left(\sum_{j=1}^n \xi_j \right) s^{\alpha' - 1} \leq \sum_{j=1}^n \xi_j s^{\alpha_j - 1} \leq \left(\sum_{j=1}^n \xi_j \right) s^{\alpha'' - 1},$$

we get

$$s' = \left(\left(\tilde{P} - \sum_{j=1}^n P_j^* \right) / \left(\frac{1}{n} \sum_{i=1}^m \tilde{\lambda}_i \bar{r}_i \right) / \left(\sum_{j=1}^n \xi_j \right) \right)^{1/(\alpha'' - 1)},$$

and

$$s'' = \left(\left(\tilde{P} - \sum_{j=1}^n P_j^* \right) / \left(\frac{1}{n} \sum_{i=1}^m \tilde{\lambda}_i \bar{r}_i \right) / \left(\sum_{j=1}^n \xi_j \right) \right)^{1/(\alpha' - 1)}.$$

For the aforementioned constant speed s and $\lambda_{i,j} \approx \lambda_i/n$, we have

$$\rho_j \approx \frac{1}{ns} \sum_{i=1}^m \tilde{\lambda}_i \bar{r}_i.$$

Since $\rho_j < 1$, we get

$$s > \frac{1}{n} \sum_{i=1}^m \tilde{\lambda}_i \bar{r}_i,$$

which can be guaranteed if

$$s' > \frac{1}{n} \sum_{i=1}^m \tilde{\lambda}_i \bar{r}_i,$$

that is,

$$\tilde{P} \left(\frac{1}{n} \sum_{i=1}^m \tilde{\lambda}_i \bar{r}_i \right)^{\alpha''} \left(\sum_{j=1}^n \xi_j \right) + \sum_{j=1}^n P_j^*.$$

TABLE 2 Numerical data for optimal load distribution

i	$\lambda_{i,1}$	$\lambda_{i,2}$	$\lambda_{i,3}$	$\lambda_{i,4}$
1	0.4760550	0.3944346	0.3363710	0.2931394
2	0.5445796	0.5127890	0.4841513	0.4584801

TABLE 3 Numerical data for optimal speed setting

i	$s_{i,1}$	$s_{i,2}$	$s_{i,3}$	$s_{i,4}$
1	1.5646660	1.4264705	1.3192646	1.2329933
2	1.6264802	1.4838095	1.3730264	1.2838000

TABLE 4 Numerical data for optimal server setting

j	ξ_j	α_j	P_j^*	ρ_j	T_j	P_j
1	1.0000000	3.0000000	2.0000000	0.7060385	1.7945432	4.8942500
2	1.2000000	3.0000000	2.5000000	0.6912183	1.8988760	5.0888871
3	1.4000000	3.0000000	3.0000000	0.6781080	1.9908953	5.3529902
4	1.6000000	3.0000000	3.5000000	0.6662990	2.0734676	5.6638727

We will let

$$\tilde{P} = \beta \left(\left(\frac{1}{n} \sum_{i=1}^m \tilde{\lambda}_i \bar{r}_i \right)^{\alpha''} \left(\sum_{j=1}^n \xi_j \right) + \sum_{j=1}^n P_j^* \right),$$

for some $\beta > 1$.

6 | A NUMERICAL EXAMPLE

Let us consider the $m = 2$ classes of applications with task arrival rates $\tilde{\lambda}_i = 1.5 + 0.5(i - 1)$, expected task execution requirements $\bar{r}_i = 1.0 + 0.2(i - 1)$, the second moments of task execution requirements $\bar{r}_i^2 = \bar{r}_i^2(1.0 + 0.2i)$, where $1 \leq i \leq m$, and $n = 4$ heterogeneous servers with coefficients $\xi_j = 1.0 + 0.2(j - 1)$ and exponents $\alpha_j = 3$ for dynamic power consumption, base power $P_j^* = 2.0 + 0.5(j - 1)$ for static power consumption, where $1 \leq j \leq n$, and power supply $\tilde{P} = 21$.

In Table 2, we show the optimal load distribution $\lambda_{i,j}$, for all $1 \leq i \leq m$ and $1 \leq j \leq n$. In Table 3, we show the optimal server speeds $s_{i,j}$, for all $1 \leq i \leq m$ and $1 \leq j \leq n$. In Table 4, we show the optimal server setting including ξ_j , α_j , P_j^* , ρ_j , T_j , and P_j , for all $1 \leq j \leq n$. All the data are generated by our algorithm in Section 4. The minimized average response time of all tasks on the n servers is $T = 1.9275173$, provided that P does not exceed \tilde{P} .

7 | SUMMARY

The problem of optimal load distribution for multiple classes of applications on heterogeneous servers with variable speeds has been addressed in this paper. Our problem is formulated as a multivariable optimization problem, ie, finding an optimal load distribution and an optimal server speed setting, such that the average task response time is minimized without exceeding certain power supply. We have studied the problem analytically by treating each server as an M/G/1 queueing system with mixed classes of tasks. We have defined a power constrained performance optimization problem and developed a numerical algorithm to solve our optimization problem by solving a system of nonlinear equations. We have also demonstrated numerical examples to show the effectiveness of our model and method. The investigation in this paper provides a rigorous treatment of the critical issue of how to provide the best QoS by consuming certain available power resource in modern data centers.

ACKNOWLEDGEMENT

The comments from 2 anonymous reviewers are acknowledged.

ORCID

Keqin Li  <http://orcid.org/0000-0001-5224-4048>

REFERENCES

1. Hamilton J. Overall data center costs. <http://perspectives.mvdirona.com/2010/09/overalldata-center-costs/>
2. Leverich JB. Future Scaling of Datacenter Power-Efficiency [PhD thesis]. Stanford, CA: Stanford University; 2014.
3. Li K. Improving multicore server performance and reducing energy consumption by workload dependent dynamic power management. *IEEE Trans Cloud Comput.* 2016;4(2):122-137.
4. Li K. Optimal speed setting for cloud servers with mixed applications. Submitted.
5. Dynamic voltage scaling. http://en.wikipedia.org/wiki/Dynamic_voltage_scaling
6. Gandhi A. Dynamic Server Provisioning for Data Center Power Management [PhD thesis]. Pittsburgh, PA: Carnegie Mellon University; 2013.
7. Ganesh L, Weatherspoon H, Marian T, Birman K. Integrated approach to data center power management. *IEEE Trans Comput.* 2013;62(6):1086-1096.
8. Pakbaznia E, Pedram M. Minimizing data center cooling and server power costs. Paper presented at: Proceedings of the 2009 ACM/IEEE International Symposium on Low Power Electronics and Design; 2009; San Francisco, CA.
9. Tuncer O, Vaidyanathan K, Gross K, Coskun AK. Coolbudget: Data center power budgeting with workload and cooling asymmetry awareness. Paper presented at: IEEE 32nd International Conference on Computer Design (ICCD); 2014; Seoul, South Korea.
10. Zapater M, Tuncer O, Ayala JL, et al. Leakage-aware cooling management for improving server energy efficiency. *IEEE Trans Parallel Distrib Syst.* 2015;26(10):2764-2777.
11. Al Sallami NM. Load balancing in green cloud computation. Paper presented at: Proceedings of the World Congress on Engineering, Vol II; 2013; London, UK.
12. Himanshi, Ahuja S. Cloud load balancing services survey and research challenges. *Int J Adv Res Comput Sci Softw Eng.* 2015;5(6):434-441.
13. Kapoor S. A survey on dynamic load balancing algorithms in cloud computing. *Adv Comput Sci Inf Technol.* 2015;2(7):87-91.
14. Katyal M, Mishra A. A comparative study of load balancing algorithms in cloud computing environment. *Int J Distrib Cloud Comput.* 2013;1(2):5-14.
15. Kaur R, Luthra P. Load balancing in cloud computing. Paper presented at: Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing; 2014; Kochi, India.
16. Khiyaita A, Bakkali HE, Zbakh M, Kettani DE. Load balancing cloud computing: State of art. Paper presented at: National Days of Network Security and Systems; 2012; Marrakech, Morocco.
17. Al Nuaimi K, Mohamed N, Al Nuaimi M, Al-Jaroodi J. A survey of load balancing in cloud computing: Challenges and algorithms. Paper presented at: International Symposium on Network Cloud Computing and Applications; 2012; London, UK.
18. Rahman M, Iqbal S, Gao J. Load balancer as a service in cloud computing. Paper presented at: IEEE 8th International Symposium on Service Oriented System Engineering; 2014; Oxford, UK.
19. Shameem PM, Shaji RS. A methodological survey on load balancing techniques in cloud computing. *Int J Eng Technol.* 2013;4(5):3801-3812.
20. Singh A, Dutta K, Gupta H. A survey on load balancing algorithms for cloud computing. *Int J Comput Appl.* 2014;6(4):66-72.
21. Anjali J, Grover M, Singh C, Singh, Sethi H. A new approach for dynamic load balancing in cloud computing. Paper presented at: National Conference on Advances in Engineering, Technology Management; 2015; Ambala, India.
22. Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S. A genetic algorithm (GA) based load balancing strategy for cloud computing. *Procedia Technol.* 2013;10:340-347.
23. Dhinesh BLD, Krishna PV. Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Appl Soft Comput.* 2013;13(5):2292-2303.
24. Gasior J, Seredynski F. Load balancing in cloud computing systems through formation of coalitions in a spatially generalized prisoner's dilemma game. Paper presented at: Third International Conference on Cloud Computing, GRIDs, and Virtualization; 2012; Nice, France.
25. Gopinath PPG, Vasudevan SK. An in-depth analysis and study of load balancing techniques in the cloud computing environment. *Procedia Comput Sci.* 2015;50:427-432.
26. Grover J, Katiyar S. Agent based dynamic load balancing in cloud computing. Paper presented at: International Conference on Human Computer Interactions; 2013; Chennai, India.
27. Li K. Optimal load distribution for multiple heterogeneous blade servers in a cloud computing environment. *J Grid Comput.* 2013;11(1):27-46.
28. Liu C, Li K, Li K. A game approach to multi-servers load balancing with load-dependent server availability consideration. *IEEE Trans Cloud Comput.* 2018. In press.
29. Sahu Y, Pateriya RK, Gupta RK. Cloud server optimization with load balancing and green computing techniques using dynamic compare and balance algorithm. Paper presented at: 5th International Conference on Computational Intelligence and Communication Networks; 2013; Mathura, India.
30. Singh A, Juneja D, Malhotra M. Autonomous agent based load balancing algorithm in cloud computing. *Procedia Comput Sci.* 2015;45:832-841.

31. Srinivasan RK, Suma V, Nedu V. An enhanced load balancing technique for efficient load distribution in cloud-based IT industries. Paper presented at: International Symposium on Intelligent Informatics; 2012; Chennai, India.
32. Tong Z, Xiao Z, Li K, Li K. Proactive scheduling in distributed computing—a reinforcement learning approach. *J Parallel Distrib Comput*. 2014;74(7):2662-2672.
33. Xiao Z, Liang P, Tong Z, Li K, Khan SU, Li K. Self-adaptation and mutual adaptation for distributed scheduling in benevolent clouds. *Concurrency Computat Pract Exper* 2017;29(5). e3939.
34. Xiao Z, Tong Z, Li K, Li K. Learning non-cooperative game for load balancing under self-interested distributed environment. *Appl Soft Comput*. 2017;52:376-386.
35. Beloglazov A, Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Futur Gener Comput Syst*. 2012;28(5):755-768.
36. Cao J, Li K, Stojmenovic I. Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers. *IEEE Trans Comput*. 2014;63(1):45-58.
37. Ghafari SM, Fazeli M, Patooghy A, Rikhtechi L. Bee-MMT: A load balancing method for power consumption management in cloud computing. Paper presented at: Sixth International Conference on Contemporary Computing; 2013; Noida, India.
38. Huang J, Li R, An J, Ntalasha D, Yang F, Li K. Energy-efficient resource utilization for heterogeneous embedded computing systems. *IEEE Trans Comput*. 2017;66(9):1518-1531.
39. Kansal NJ, Chana I. Cloud load balancing techniques: a step towards green computing. *Int J Comput Sci Issues*. 2012;9(1):238-246.
40. Li K. Optimal task dispatching for multiple heterogeneous multiserver systems with dynamic speed and power management. *IEEE Trans Sustain Comput*. 2017;2(2):167-182.
41. Malik SUR, Bilal K, Khan SU, Veeravalli B, Li K, Zomaya AY. Modeling and analysis of the thermal properties exhibited by cyber physical data centers. *IEEE Syst J*. 2017;11(1):163-172.
42. Paul D, Zhong W-D, Bose SK. Energy efficiency aware load distribution and electricity cost volatility control for cloud service providers. *J Netw Comput Appl*. 2016;59:185-197.
43. Tian Y, Lin C, Li K. Managing performance and power consumption tradeoff for multiple heterogeneous servers in cloud computing. *Clust Comput*. 2014;17(3):943-955.
44. Yang B, Li Z, Chen S, Wang T, Li K. A stackelberg game approach for energy-aware resource allocation in data centers. *IEEE Trans Parallel Distrib Syst*. 2016;27(12):3646-3658.
45. Kleinrock L. *Queueing Systems*, Vol. 1: Theory. New York, NY: John Wiley and Sons; 1975.
46. Burden RL, Faires JD, Reynolds AC. *Numerical Analysis*. 2nd ed. Boston, MA: Prindle, Weber & Schmidt; 1981.

How to cite this article: Li K. Optimal load distribution for multiple classes of applications on heterogeneous servers with variable speeds. *Softw Pract Exper*. 2018;1–15. <https://doi.org/10.1002/spe.2584>