Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 594264, 13 pages
http://dx.doi.org/10.1155/2015/594264

Research Article

Modeling and Analysis of Queueing-Based Vary-On/Vary-Off Schemes for Server Clusters

Cheng-Jen Tang[1] and Miau-Ru Dai[2]

[1]*Department of Electrical Engineering, Tatung University, Taipei 10452, Taiwan*
[2]*Delta Network Inc., Taipei 11491, Taiwan*

Correspondence should be addressed to Cheng-Jen Tang; ctang@ttu.edu.tw

Received 7 January 2015; Accepted 30 May 2015

Academic Editor: Stefano de Miranda

A cloud system usually consists of a lot of server clusters handling various applications. To satisfy the increasing demands, especially for the front-end web applications, the computing capacity of a cloud system is often allocated for the peak demand. Such installation causes resource underutilization during the off-peak hours. Vary-On/Vary-Off (VOVO) schemes concentrate workloads on some servers instead of distributing them across all servers in a cluster to reduce idle energy waste. Recent VOVO schemes adopt queueing theory to model the arrival process and the service process for determining the number of powered-on servers. For the arrival process, Poisson process can be safely assumed in web services due to the large number of independent sources. On the other hand, the heavy-tailed distribution of service times is observed in real web systems. However, there are no exact solutions to determine the performance for *M/heavy-tailed/m* queues. Therefore, this paper presents two queueing-based sizing approximations for Poisson and non-Poisson governed service processes. The simulation results of the proposed approximations are analyzed and evaluated by comparing with the simulated system running at full capacity. This relative measurement indicates that the Pareto distributed service process may be adequately modeled by memoryless queues when VOVO schemes are adopted.

1. Introduction

The numbers of Internet requests are not uniformly distributed over time. There are a huge number of requests during the peak hours. Cloud service providers tend to install surplus server nodes to handle the bursty load. Clearly, these servers waste a lot of energy during the off-peak periods. Dynamically adjusting the number of active servers, that is, Vary-On/Vary-Off (VOVO) scheme, improves energy-efficiency of server clusters. However, overly shrinking the number of powered-on servers may lead to decreased service quality. Therefore, finding the *right* number of active servers to balance energy consumption and operation performance is a primary issue of an applicable VOVO scheme.

VOVO schemes can be dated back to earlier last decade [1, 2]. The basic idea of earlier VOVO schemes is to dynamically size a cluster according to CPU utilization or resource usage. This resource provisioning problem in a cluster can be analogous to the staff sizing problem in a telephone call center. In a call center, the customers are callers, servers are telephone agents, and tele-queues consist of callers that await service by an agent. The well-known *Erlang-C* model [3] has been widely applied to this problem. Many recent VOVO studies [4–9] adopt queueing analysis to manage resource usage of clusters.

Most available analytic solutions in queuing theory rely on independence assumptions and Poisson processes [10]. Internet traffic patterns are well known to possess extreme variability and bursty structure [11]. The heavy-tailed distributions of service times are observed in real web systems [12, 13]. This characteristic is characterized by self-similar process [14]. Pareto distribution is a popular model of self-similar processes [15]. However, queueing models with Pareto distributed service times are very difficult to analyze [16]. Although heavy-tailed service processes in web systems are widely documented, memoryless queues are still used for evaluating system performance in many studies [17–21]. On the other hand, studies [22–25] that adopt general/Pareto distributions need approximations for the analytically intractable distributions to obtain the performance measures.

The Poisson arrival process is particularly appropriate if the arrivals are from a large number of independent sources [10], such as users of web services. However, exploring the difference between modeling service times with Poisson process and non-Poisson process governed queues remains a challenging research topic, since many queueing models remain analytically intractable [26]. In order to understand the performance difference between modeling service times with Poisson process and non-Poisson process governed queues, a series of simulations are conducted in this study. Compared with the mathematical analysis and numerical methods, simulation is more time and memory consuming but it is sometimes the only way to get reasonably accurate results [27].

This paper presents the approximations of VOVO cluster sizing for systems modeled by $M/G/m$ and $M/M/m$. Randomly generated workload traces with Pareto and exponential distributed service times are simulated using the approximations. Two distinct types of real web access logs are simulated as well. A relative performance evaluation method is proposed and used for gauging the simulation results. Through the evaluation, the performance difference between modeling service times with Poisson process and non-Poisson process governed queues is found. The result suggests that $M/M/m$ based sizing approach may be adequate when a queueing-based VOVO scheme is adopted in a cluster.

This paper is organized as follows. Section 2 shows the approximation methods for cluster sizing. Section 3 details the simulation setup and the evaluation metric. Section 4 presents the simulation process and discusses the results. Section 5 concludes this paper.

2. Approximation for Queueing-Based Cluster Sizing

Investigations in a queueing theory applied system mainly aim at getting the performance measures, which are the probabilistic properties of the random variables, including number of customers in the system, number of waiting customers, utilization of the servers, response time of a request, waiting time of a customer, idle time of the server, and busy time of a server. These measures heavily depend on the assumptions concerning the distributions of interarrival times and service times as well as number of servers and service discipline. Queueing analysis can be naturally applied to the performance measures of server clusters. Server clusters have been widely adopted in many cloud data centers to resolve the increasing user needs [28]. Although heterogeneity is common in multifunctional cloud data centers, server closets or blade systems that form the basic computing units usually consist of homogeneous nodes. Therefore, this work focuses on single-queue homogeneous systems.

The symbols and definitions used in this paper to describe the performance measures of queueing systems are shown in Symbols and Definitions.

In classical queueing analysis, supposing that requests are handled by a single-queue m homogeneous server system with the First-Come First-Serve (FCFS) discipline, exponentially distributed service times, and Poisson process governed arrival intervals, the system can be modeled as $M/M/m$ system. ρ must be less than 1 ($\rho < 1$) for $M/M/m$ system being in a stable state. Many performance measures of a stable $M/M/m$ system have been thoroughly studied and are shown in (1) to (7). The calculations and proofs of these equations can be found in many textbooks, for example, [29, p. 412]:

$$p_j = \begin{cases} p_0 \dfrac{(m\rho)^j}{j!}, & \text{if } 0 < j < m \\ p_0 \dfrac{(m\rho)^j}{m! m^{j-m}}, & \text{if } j \geq m, \end{cases} \tag{1}$$

$$p_0 = \left(\sum_{j=0}^{m-1} \frac{(m\rho)^j}{j!} + \frac{(m\rho)^m}{m!(1-\rho)} \right)^{-1}, \tag{2}$$

$$E[N] = m\rho + \frac{\rho p_m}{(1-\rho)^2}, \tag{3}$$

$$E[M] = \frac{\lambda}{\mu}, \tag{4}$$

$$\begin{aligned} E[R] &= \frac{E[N]}{\lambda} = \frac{1}{\mu} + \frac{p_m}{m\mu(1-\rho)^2} \\ &= \frac{1}{\mu} + \frac{p_m}{m\mu(1-\lambda/m\mu)^2}, \end{aligned} \tag{5}$$

$$E[N_1] = \frac{\rho}{1-\rho}, \tag{6}$$

$$E[R_1] = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu(1-\lambda/\mu)}. \tag{7}$$

2.1. Approximation for Sizing $M/M/m$ Modeled Clusters. In a homogeneous $M/M/m$ system, μ of each server is identical. From (5), $E[R]$ of $M/M/m$ system can be considered as a function of λ denoted by $f_m(\lambda)$:

$$f_m(\lambda) = \frac{1}{\mu} + \frac{p_m}{m\mu(1-\lambda/m\mu)^2}. \tag{8}$$

Let λ_m be an arrival rate of $M/M/m$ system maintaining a targeted response time r given $r > 1/\mu$. The curves of $f_m(\lambda)$ for $m = k-2$, $m = k-1$, $m = k$, $m = k+1$, and $m = k+2$ with a targeted response time r are shown in Figure 1.

λ_1 can be easily obtained from (5):

$$r = \frac{1}{\mu(1-\lambda_1/\mu)}, \tag{9}$$

$$\lambda_1 = \mu - \frac{1}{r}. \tag{10}$$

For $m > 1$, r, based on (7), can be represented as

$$r = \frac{1}{\mu} + \frac{p_m}{m\mu(1-\lambda_m/m\mu)^2}. \tag{11}$$

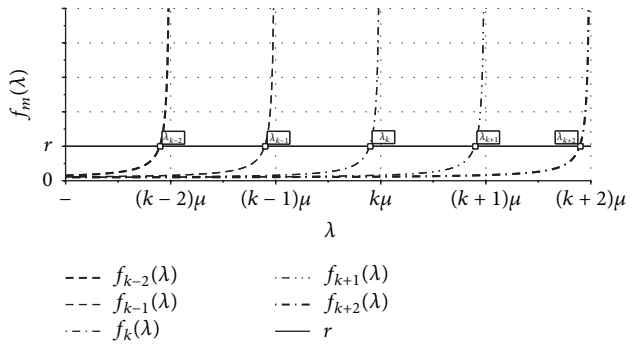


FIGURE 1: $f_m(\lambda)$ versus arrival rate (λ) with a targeted response time r.

From (1) and (2), p_m is

$$\begin{aligned} p_m &= p_0 \frac{\left(m\left(\lambda_m/m\mu\right)\right)^m}{m!} \\ &= \frac{\left(\lambda_m/\mu\right)^m/m!}{\left(\sum_{j=0}^{m-1}\left(\left(\lambda_m/\mu\right)^j/j!\right)+\left(\lambda_m/\mu\right)^m/m!\left(1-\lambda_m/m\mu\right)\right)}. \end{aligned} \tag{12}$$

Therefore, to get λ_m of μ for $m \geq 2$, the following equation has to be solved:

$$r = \frac{1}{\mu} + \frac{\left(\lambda_m/\mu\right)^m}{m!m\mu\left(\sum_{j=0}^{m-1}\left(\left(\lambda_m/\mu\right)^j/j!\right)+\left(\lambda_m/\mu\right)^m/m!\left(1-\lambda_m/m\mu\right)\right)\left(1-\lambda_m/m\mu\right)^2}. \tag{13}$$

λ_2 can also be easily obtained by solving (13) with $m = 2$:

$$\lambda_2 = 2\sqrt{\mu^2 - \frac{\mu}{r}}. \tag{14}$$

It is difficult to get a closed-form expression of λ_m in terms of r, μ, and m when $m > 2$. Therefore, an approximation is proposed for λ_m for $m > 2$. Assume that this approximation can be applicable for the systems with at most c servers. Every $f_m(\lambda)$, $c \geq m \geq 2$, is shifted with the offset value of $-(m-1)\mu$ and denoted as $g_m(\lambda)$:

$$g_m(\lambda) = f_m\left(\lambda + (m-1)\mu\right), \quad \text{for } c \geq m \geq 2. \tag{15}$$

Figure 2 shows the combination of the curves of $f_1(\lambda)$ and $g_m(\lambda)$, $c \geq m \geq 2$, with emphasis on the intersections between the targeted response time r and these curves.

By observing Figure 2, the distances between all consecutive $\lambda_{m-1} - (m-2)\mu$ and $\lambda_m - (m-1)\mu$ approximately form an exponential decay series $\{\delta_1, \delta_2, \ldots, \delta_c\}$. Let the series be approximated by an exponential decay function, let α be the initial quantity, and let β be the exponential decay constant. An element δ_m in the series can be expressed as

$$\delta_m = \alpha e^{-\beta(m-1)} = \lambda_{m-1} - \lambda_m + \mu. \tag{16}$$

Let the initial quantity $\alpha = \delta_1$; α can be obtained from (10):

$$\alpha = \mu - \lambda_1 = \mu - \left(\mu - \frac{1}{r}\right) = \frac{1}{r}. \tag{17}$$

From (17), (16), (10), and (14), δ_2 is

$$\delta_2 = \lambda_1 - \lambda_2 + \mu = 2\mu - \frac{1}{r} - 2\sqrt{\mu^2 - \frac{\mu}{r}} = \frac{1}{r}e^{-\beta}. \tag{18}$$

β can be obtained by rearranging (18):

$$\beta = -\ln\left(2\mu r - 2\sqrt{\mu^2 r^2 - \mu r} - 1\right). \tag{19}$$

Therefore, δ_m can be represented as

$$\delta_m = \frac{1}{r}\left(2\mu r - 2\sqrt{\mu^2 r^2 - \mu r} - 1\right)^{(m-1)}. \tag{20}$$

Let $\theta = (2\mu r - 2\sqrt{\mu^2 r^2 - \mu r} - 1)$. For a positive integer $m \geq 1$, λ_m can be approximated as

$$\lambda_m = m\mu - \sum_{i=1}^{m}\delta_m = m\mu - \frac{1-\theta^m}{r\left(1-\theta\right)}. \tag{21}$$

Consequently, with an anticipated arrival rate λ and the measured service rate μ, the number, denoted as m, of servers that maintain the targeted mean response time r can be approximated as

$$m = \begin{cases} \left\lceil \dfrac{\lambda}{\mu} \right\rceil, & \text{if } \left(\mu\left\lceil \dfrac{\lambda}{\mu} \right\rceil - \dfrac{1-\theta^{\lceil \lambda/\mu \rceil}}{r\left(1-\theta\right)}\right) > \lambda \\ \left\lceil \dfrac{\lambda}{\mu} \right\rceil + 1, & \text{otherwise.} \end{cases} \tag{22}$$

2.2. Approximation for Sizing $M/G/m$ Modeled Clusters. Internet workload characterization has found that the probability of service times is not an exponential distribution but a heavy-tailed distribution in real web systems [12, 14, 30]. In other words, a single-queue m-server cluster should be referred to as $M/G/m$ queue for Internet services. There are no exact formulas for the mean response time of $M/G/m$ system, but numerous approximations can be used. *Kingman's Exponential Law of Congestion* is a popular approximation that is calculated using the coefficient of variation of service times and known solutions from $M/M/m$ queues. Kingman's approximation is expressed as

$$E\left[W^+\right] \approx \frac{1+C^2}{2}E\left[W\right]. \tag{23}$$

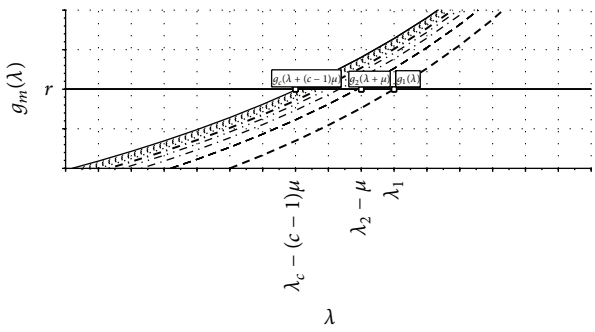


FIGURE 2: $g_m(\lambda)$ versus request arrival rate (λ) with a targeted response time r.

Let $\nu = (1+C^2)/2$. The mean response time of $M/G/m$ system can be expressed as

$$E\left[R^+\right] = \frac{1}{\mu} + E\left[W^+\right] = \frac{1}{\mu} + \nu\frac{p_m}{m\mu\left(1-\lambda/m\mu\right)^2}. \quad (24)$$

Let $f_m^+(\lambda)$ represent the mean response time of $M/G/m$ system on different arrival rates. Based on (8), $f_m^+(\lambda)$ can be expressed as

$$f_m^+(\lambda) = f_m(\lambda) + (\nu-1)\frac{p_m}{m\mu\left(1-\lambda/m\mu\right)^2}. \quad (25)$$

Although $f_m^+(\lambda)$ rises at a more precipitous rate than $f_m(\lambda)$, the correlation observed in Figure 2 and aforementioned approximation still remain valid.

Let the variables $\{\lambda_1^+, \lambda_2^+, \ldots, \lambda_c^+\}$, $\{\delta_1^+, \delta_2^+, \ldots, \delta_c^+\}$, and θ^+ be the correspondences in $M/G/m$ model to the variables $\{\lambda_1, \lambda_2, \ldots, \lambda_c\}$, $\{\delta_1, \delta_2, \ldots, \delta_c\}$, and θ previously mentioned in $M/M/m$ model. The mean response time for $M/G/1$ system can be approximated based on the *Pollaczek-Khintchine* transform:

$$E\left[R_1^+\right] \approx \frac{1}{\mu} + \frac{\nu\rho}{\mu\left(1-\rho\right)} = f_1^+\left(\lambda_1^+\right). \quad (26)$$

Suppose that the targeted response time is still r; then

$$r = f_1^+\left(\lambda_1^+\right) = \frac{1}{\mu} + \nu\frac{\lambda_1^+/\mu}{\mu\left(1-\lambda_1^+/\mu\right)},$$
$$\lambda_1^+ = \frac{\mu\left(r\mu-1\right)}{r\mu-1+\nu}. \quad (27)$$

Similar to the process from (14) to (20), the following equations can be derived:

$$\lambda_2^+ = 2\mu\sqrt{\frac{r\mu-1}{r\mu-1+\nu}},$$
$$\delta_1^+ = \mu - \lambda_1^+ = \frac{\mu\nu}{r\mu-1+\nu},$$
$$\delta_2^+ = \frac{2\mu\left(r\mu-1\right)+\mu\nu}{r\mu-1+\nu} - 2\mu\sqrt{\frac{r\mu-1}{r\mu-1+\nu}},$$
$$\theta^+ = \frac{2\left(r\mu-1\right)+\nu-2\sqrt{\left(r\mu-1\right)\left(r\mu-1+\nu\right)}}{\nu},$$
$$\lambda_m^+ = m\mu - \frac{1-\left(\theta^+\right)^m}{r\left(1-\theta^+\right)}. \quad (28)$$

With an anticipated arrival rate λ and the measured service rate μ, the number, denoted as m^+, of servers that is expected to maintain the required mean response time r can be approximated as

$$m^+ = \begin{cases} \left\lceil \frac{\lambda}{\mu} \right\rceil, & \text{if } \left(\mu\left\lceil \frac{\lambda}{\mu} \right\rceil - \frac{1-\left(\theta^+\right)^{\lceil\lambda/\mu\rceil}}{r\left(1-\theta^+\right)}\right) > \lambda \\ \left\lceil \frac{\lambda}{\mu} \right\rceil + 1, & \text{otherwise.} \end{cases} \quad (29)$$

3. Simulation Setup and Evaluation Metric

A cluster managed by a VOVO scheme periodically adjusts the number of active servers that provide the required services. In general, there are several key functional components including the following:

(1) Job queue: the job queue holds the waiting requests. Each request enters the tail of the queue and waits for service in FCFS manner. In this work, all jobs share a common queue.

(2) Workload distributor: the workload distributor retrieves a job from the head of the job queue and distributes the job to an available node.

(3) Cluster sizing unit: this unit decides the number of active servers. The decision may be based on some predefined thresholds of certain resources, for example, CPU utilization, job throughput, and energy usage. In this work, the decision is calculated based on (22) or (29) according to the given arrival rate, mean service rate, and targeted response time.

(4) On/off controller: the on/off controller periodically activates or deactivates server nodes according to the number given by the sizing unit.

(5) Managed servers: the cluster consists of a group of identical computer nodes, which may be commodity servers. Each server node processes the assigned jobs and reports its working status to the workload distributor.

3.1. The Design of Simulation Program. A simulation program for the VOVO managed system is developed to investigate the performance of the proposed sizing methods. This program is written using the C++ programming language.

In a real VOVO managed system, every incoming job is queued, and an event notification is issued to the workload

```
Simulation(queue, control_interval, state)
cur_period ← −1
next_period ← −1
while queue is not empty do
   Retrieve the job in the head of queue and remove it from queue
   if current_period < 0 then
      cur_period ← job.arrival_time − mod(job.arrival_time, control_interval)
      next_period ← cur_period + control_interval
   else if job.arrival_time ≥ next_period then
      while job.arrival_time ≥ next_period do
         perform statistic of the activated server nodes
         perform cluster sizing
         do on-off control
         cur_period ← next_period
         next_period ← cur_period + control_interval
      end while
   end if
   find an activated server node with the earliest available time
   calculate the waiting time of this job
   calculate the service time of this job
   calculate the response time of this job
   calculate the consumed energy of the server node
end while
```

Algorithm 1: Simulation process.

distributor upon the arrival of a job. If there are available nodes, the workload distributor then dispatches the queued jobs to the available nodes. If a node has completed its assigned job, it also sends an event notification to inform the distributor about its availability. The instructions of node activation and deactivation are periodically issued by the on/off controller. If a deactivation command is issued to a busy node, the node will complete the processing job before turning itself off. However, it will be extremely time consuming to simulate the system with time-based event-driven process. Since the input workload traces have to be readily prepared for the simulation, this work adopts the sequential process that significantly reduces the simulation time. The simulation process is shown in Algorithm 1.

3.2. Randomly Generated Traces. A set of randomly generated traces and two real-world traces are simulated in this work. The most widely used heavy-tailed distribution as the service time distribution is the Pareto distribution [31]. The Poisson distribution is appropriate if the arrivals are from a large number of independent sources, such as web requests [10, 32]. Therefore, the randomly generated traces have Pareto distributed service times with tail indexes from 0.1 to 4.0 stepping by 0.1 and exponentially distributed arrival intervals with traffic intensities from 0.05 to 0.95 stepping by 0.05.

A randomly generated trace T with a tail index l and a traffic intensity ρ is represented by $T_{l,\rho}$. $T_{l,\rho}$ is a series of pairs of an arrival time, denoted by t_a, and a service time, denoted by t_s. Suppose that $T_{l,\rho}$ has n elements; it can be represented as

$$T_{l,\rho} = \{(t_{a1}, t_{s1}), (t_{a2}, t_{s2}), \ldots, (t_{an}, t_{sn})\}. \quad (30)$$

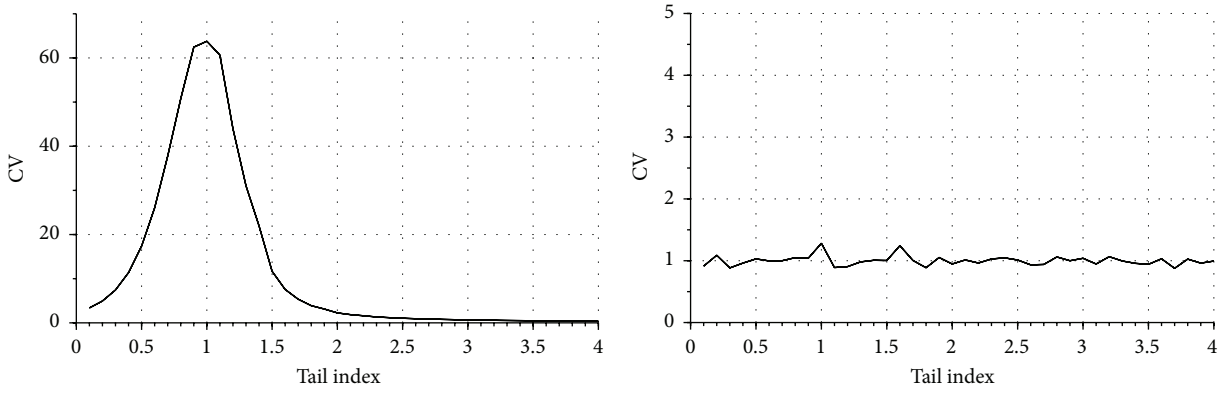
Each unique combination of l and ρ is randomly generated 10 times. That is, there are 10 different traces for a combination of l and ρ. Each trace contains values covering 36,000 time units. All traces are generated with the same mean service time. Therefore, there are 7,600 randomly generated traces which have been simulated in this study. The generating functions for Pareto distributed values and exponential distributed values can be found in many textbooks, for example, [10, p. 509]. The coefficient of variation is often used to measure the relative variation in the data and is the ratio of the standard deviation to the mean. For Pareto distributed values, the coefficient of variation denoted by $\mathrm{CV}_{\mathrm{Pareto}}$ of $l > 2$ can be calculated as [10]

$$\mathrm{CV}_{\mathrm{Pareto}} = \frac{\sqrt{l/(l-1)^2(l-2)}}{l/(l-1)} = \sqrt{\frac{1}{l(l-2)}}, \quad \text{for } l > 2. \quad (31)$$

The coefficient of variation of exponential distributed values is supposed to be 1. The coefficients of variation of service times and arrival intervals of the generated traces are shown in Figures 3(a) and 3(b), respectively.

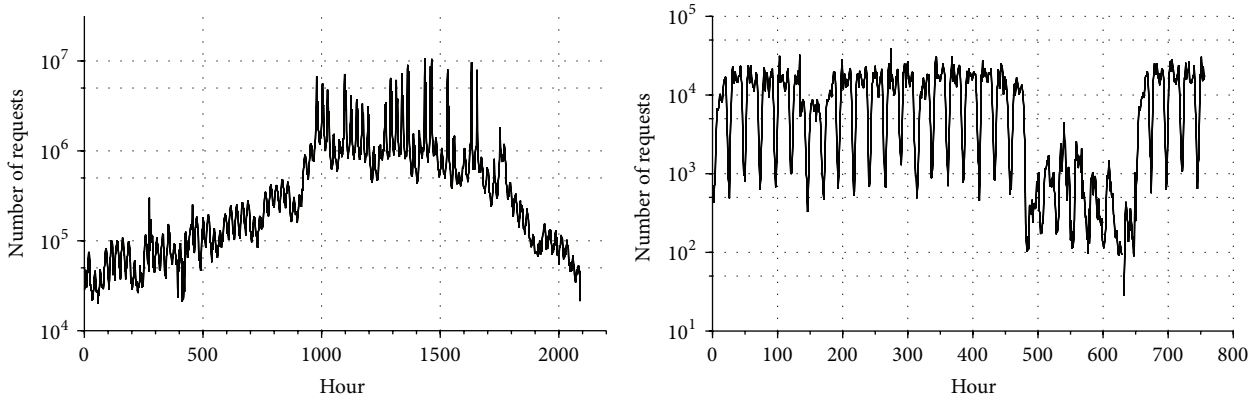
3.3. Real-World Traces. This simulation adopts two real-world workload traces that include a publicly available trace and a trace acquired from a university campus. The service time of a request is assumed to be proportional to its responded page size in the simulation.

The publicly available trace was recorded at the 1998 World Cup web site [30]. This workload trace is one of few



(a) CV of service times versus tail index (b) CV of arrival intervals versus tail index

Figure 3: The coefficients of variation (CV) of service times and arrival intervals of the generated traces.



(a) Traffic pattern of the World Cup log (b) Traffic pattern of the campus log

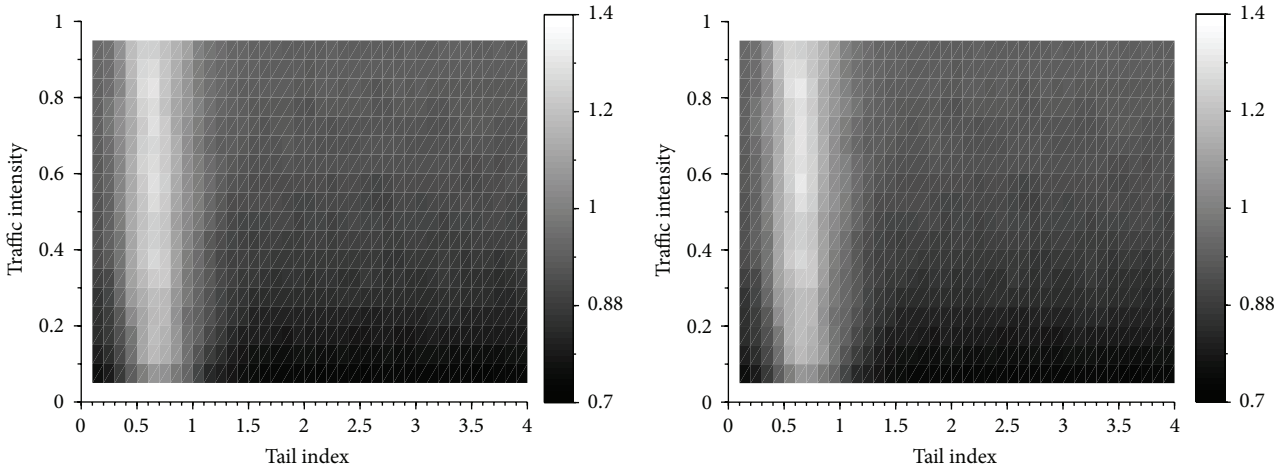
Figure 4: Hourly traffic patterns of the adopted real-world web traces.

logs providing server activation records. It is known for having a heavy-tailed page-size distribution with a tail index of 1.37 [30]. Each request recorded in the log contains an arrival time, a responded page size, and a server identification. The 1998 World Cup log was collected from 05:30:17 May 1, 1998, through 05:59:55 July 27, 1998, a total of 87 days. The log exhibits the following characteristics: 1,352,804,107 requests, 33 hosting servers, 4,040.684 bytes per response in average, 108.71 requests per second per server (the peak service rate) [30], and an average service time of 0.0092 seconds per request with a standard deviation of 0.084.

The second workload trace is acquired from a university with a student population of 4,219, including 3,531 undergraduates. This web access log was collected from 12:03:59 September 19, 2014, through 00:01:39 October 21, 2014, a total of 31 days. The trace log is from a site hosting a student information system that provides course information, handouts/homework systems, message system, email system, and other campus information. The log exhibits the following characteristics: 7,054,170 requests, 8 hosting servers, 5,991.64 bytes per response in average, 74.74 requests per second per server (the peak service rate), an average service time of 0.0134 seconds per request with a standard deviation of 0.227, and a tail index of 0.154 of the service time distribution.

The hourly traffic patterns of the 1998 World Cup log and the 2014 campus log are shown in Figures 4(a) and 4(b), respectively. The two logs represent two distinct service patterns including an occasional service pattern, that is, 1998 World Cup, and a regular service pattern, that is, student information system. The World Cup log shows a growth-decay pattern. An iterative pattern analogous to the daily working hours is observed in the campus log. Note that there are a school break and a scheduled maintenance during the recorded period.

For the World Cup log, the simulated cluster consists of 33 servers based on the information given in the log. For the campus log, the simulated cluster consists of 8 servers. As for the randomly generated traces, the simulated cluster consists of 10 servers. The on/off controller periodically sizes the simulated cluster with the interval set at 300 seconds, which are long enough to compensate the machine boot-up delays and short enough to reflect the demand changes [1, 2, 33].



(a) Relative performance of *mgm* scenario

(b) Relatively performance of *mmm* scenario

FIGURE 5: Relative performance of the randomly generated traces with $\omega_S = 1$, $\omega_A = 1$, and $\omega_E = 1$.

3.4. Evaluation Metric. Three simulation scenarios, which are all-on, *mmm*, and *mgm*, are performed. All servers in a cluster are always powered on in all-on scenario. This scenario is expected to consume the most energy but to have the best service quality. The *mmm* scenario uses (22) to approximate the number of servers. The *mgm* scenario is similar to *mmm* except that (29) is used for the sizing approximation. Nielsen's [34] response time limits for usability are adopted by setting the targeted response time at 1 second and the failure threshold at 10 seconds.

The objective of a VOVO scheme is to reduce the energy consumption while maintaining a reasonable service quality. To gauge the performance of an approach x (denoted by η_x), relative measures to all-on are adopted instead of absolute measurements, since the all-on scenario must have the least response time and the highest energy consumption. The considered factors of a scenario x are as follows:

(1) being satisfactory, denoted by S_x, which is the portion of responses conforming to the targeted response time;

(2) acceptance, denoted by A_x, which is the portion of responses being admissible (i.e., under the failure threshold);

(3) energy, denoted by E_x, which is the average number of activated servers, since all servers are identical and have the same power profile.

The relative measurements of S_x, A_x, and E_x are defined as

$$S_x^+ = \frac{S_{\text{all-on}}}{S_x}, \quad \text{for } S_x > 0,$$
$$A_x^+ = \frac{A_{\text{all-on}}}{A_x}, \quad \text{for } A_x > 0, \tag{32}$$
$$E_x^+ = \frac{E_x}{E_{\text{all-on}}}, \quad \text{for } E_{\text{all-on}} > 0.$$

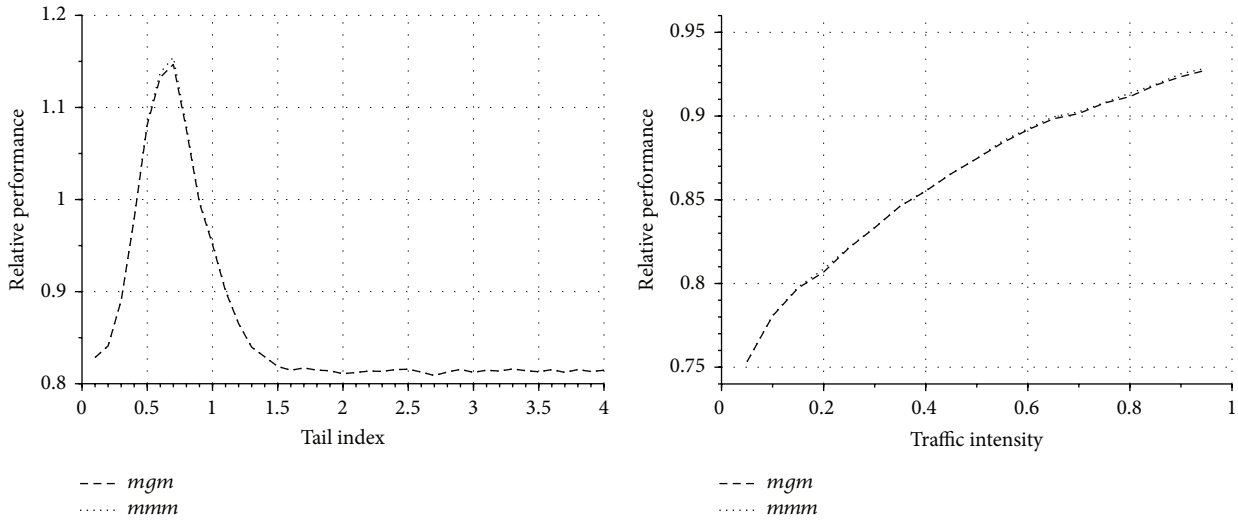
Let ω_S, ω_A, and ω_E be the weighting coefficients for S_x^+, A_x^+, and E_x^+, respectively. The relative performance, denoted by η_x, is defined as

$$\eta_x = \frac{\left(S_x^+\omega_S + A_x^+\omega_A + E_x^+\omega_E\right)}{\left(\omega_S + \omega_A + \omega_E\right)}. \tag{33}$$

4. Simulation Results and Analysis

4.1. Simulation Results. With this relative measurement, that is, (33), the optimal solution produces the minimal value of η_x. The simulation results of randomly generated traces are summarized by the relative performance of the simulated scenarios to all-on with $\omega_S = 1$, $\omega_A = 1$, and $\omega_E = 1$. In order to make the results be easily comprehended, the relative performances of η_{mgm} and η_{mmm} are graphically visualized using gray level. Figure 5 shows the relative performance, that is, η_x, of scenarios *mgm* and *mmm*, with $\omega_S = 1$, $\omega_A = 1$, and $\omega_E = 1$. It is very difficult to visually differentiate Figures 5(a) and 5(b). Using the averaged values, as shown in Figure 6, it can be found that *mgm* has a slightly better performance than *mmm*. In average, which is based on Figure 6, scenarios *mgm* and *mmm* outperform all-on under most cases except when the tail index is between 0.4 and 0.9. Furthermore, the averaged relative performances shown in Figure 6(a) are clearly correlated with the coefficient of variation of service times (as shown in Figure 3(a)). This simulation result indicates that both *mgm* and *mmm* yield a worse performance than all-on for diverse access patterns. This may imply that these approaches undersize the cluster for high variation of service times.

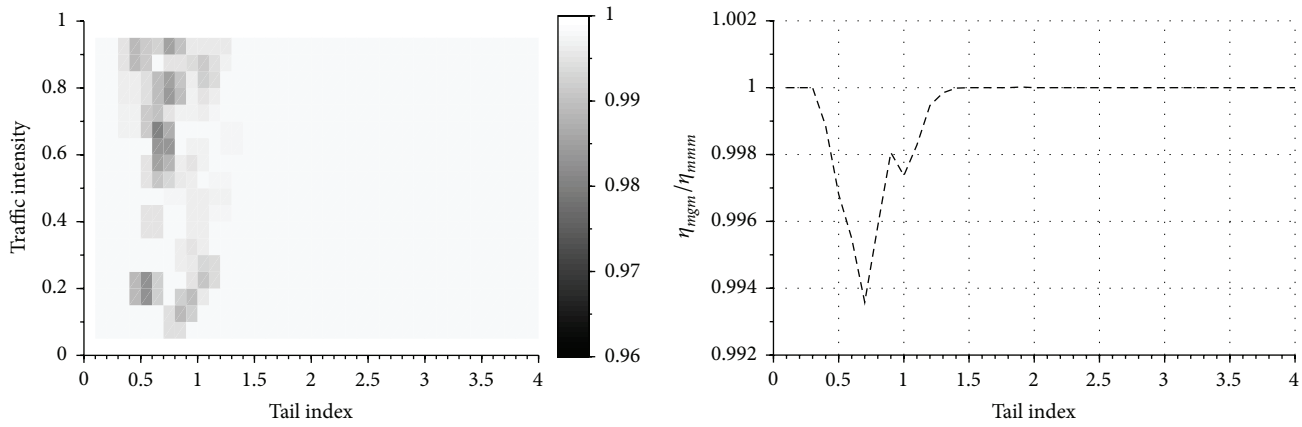
In Figures 6(a) and 6(b), the curves of *mgm* and *mmm* are indistinguishable under those scales. In fact, the relative performances of scenarios *mgm* and *mmm* are not identical. Figure 7(a) shows the ratios of η_{mgm} to η_{mmm}. There are some regions between tail indexes 0.3 and 1.3 where the ratios are not 1, that is, identical. In Figure 7(b), the average of η_{mgm}/η_{mmm} is always less than or equal to 1, which means that



(a) Relative performances versus tail index

(b) Relative performances versus traffic intensity

Figure 6: Average relative performance of the randomly generated traces with $\omega_S = 1$, $\omega_A = 1$, and $\omega_E = 3$.



(a) Relative performance ratios of *mgm* to *mmm*

(b) Average of relative performance ratios

Figure 7: The relative performance ratios of η_{mgm}/η_{mmm} with $\omega_S = 1$, $\omega_A = 1$, and $\omega_E = 1$.

$M/G/m$ based sizing is more effective than $M/M/m$ based sizing. However, Figure 7 also shows that the difference is very small, that is, under 1% in average. Given the fluctuation nature of web traffic, $M/M/m$ based sizing may be adequate for empirical practices.

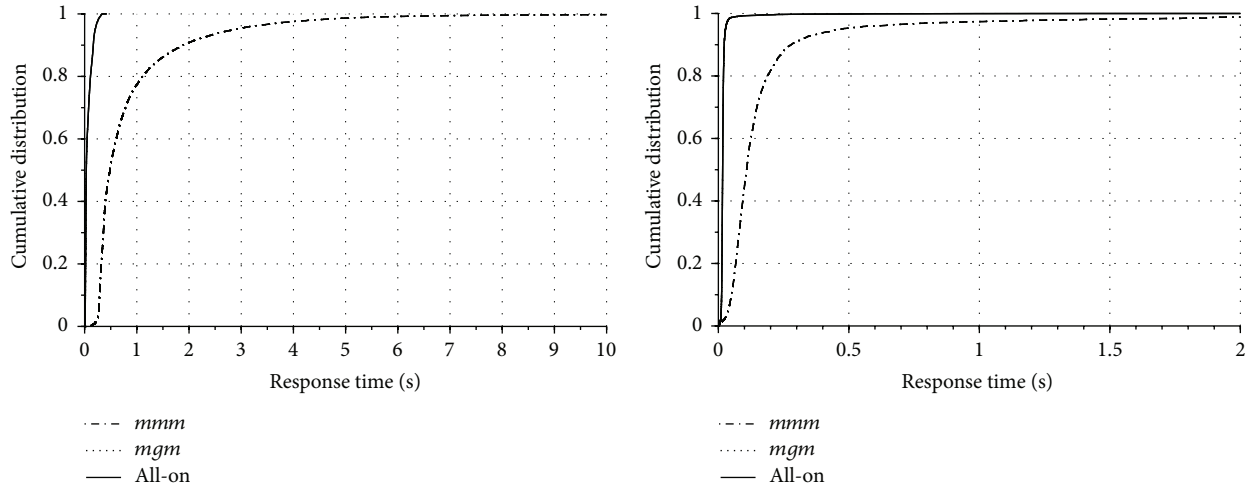
In order to examine above findings, two real-world traces are simulated under previously mentioned scenarios, that is, all-on, *mgm*, and *mmm*. Figure 8 shows the cumulative distribution of the response times of the simulated real-world traces. As shown in Figure 8(a), all requests in scenario all-on can be served within 1 second, but only approximately 80% of requests can be handled for this targeted response time in scenarios *mgm* and *mmm*. The curves of *mgm* and *mmm* are also indistinguishable in Figure 8(a). In Figure 8(b), more than 99.96% of requests in scenario all-on can be served within 1 second. More than 97% of requests can be handled for this targeted response time in *mgm* and *mgm* scenarios. The curves of *mgm* and *mmm* are also indistinguishable in Figure 8(b).

Based on the relative performance, that is, η_x, Table 1 shows that *mmm* and *mgm* are very similar in both cases. As expected, all-on always has the shortest mean response time but the most energy consumption. The proposed queueing-based sizing approaches, that is, *mmm* and *mgm*, can reduce significant energy consumption while maintaining a reasonable service quality.

4.2. Analysis and Comparison. Energy consumption and service quality of the server machines are two major performance measures for a cloud service provider. The above results are fully based on simulation. To evaluate the proposed strategy on a real system, a 6-hour log is extracted from the World Cup trace and fed to a cluster consisting of 33 computers. In addition to the 33-node cluster, there

Table 1: Relative performance.

Log	x	S_x	A_x	E_x	S_x^+	A_x^+	E_x^+	η_x
World Cup	All-on	1.000	1.000	33	1.000	1.000	1.000	1.000
	mmm	0.817	0.993	2.389	1.224	1.007	0.072	0.768
	mgm	0.817	0.993	2.390	1.223	1.007	0.072	0.767
Campus	All-on	0.999	0.999	10	1.000	1.000	1.000	1.000
	mmm	0.974	0.998	1.32	1.026	1.002	0.132	0.720
	mgm	0.974	0.998	1.32	1.026	1.002	0.132	0.720



(a) CDF of response times (World Cup)

(b) CDF of response times (campus)

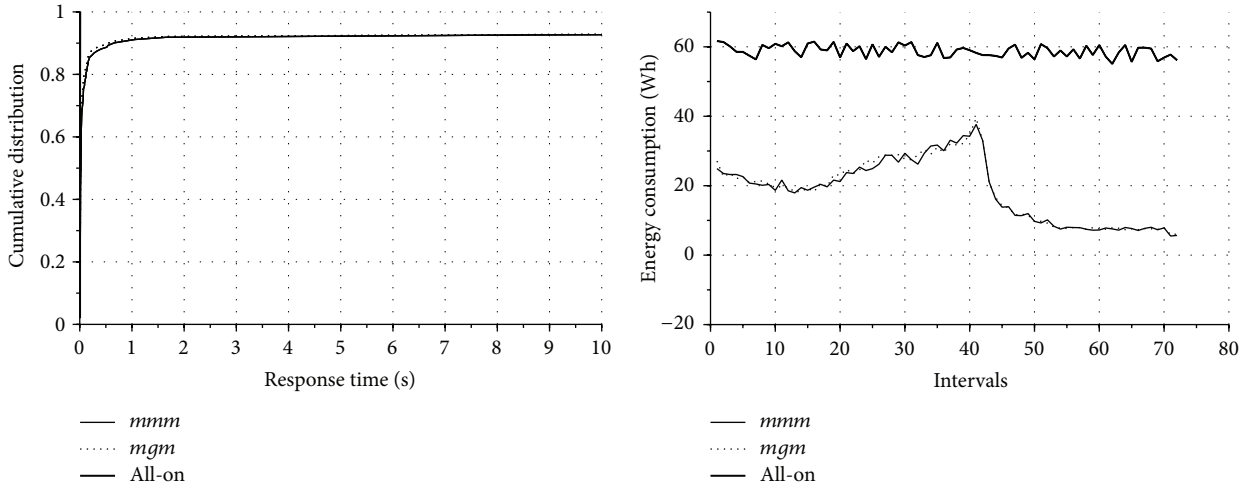
Figure 8: Cumulative distribution of the response times of two real-world traces.

are an external computer that hosts other key functional components mentioned in Section 3 and a network switch connecting all nodes and the external computer through 1000BASE-T Ethernet. The extracted log contains 22,821,177 access records, which are from June 29, 1998, 17:20:00 GMT to June 29, 1998, 23:19:59 GMT. Each node of the cluster is equipped with a dual-core 1.66 GHz Intel Atom N280 processor and 1 GB of memory. All nodes use Linux 2.6 as the operation system with Apache 2.2 installed. The average power demand is 20.83 Watts when an idle node waits for a request with all its parts being turned on. The peak power level of a node that was instrumented is 26.33 Watts. The node profile of the test cluster is shown in Table 2.

In the evaluation, the on/off controller periodically sizes the cluster with the interval set at 300 seconds. Interval energy data of the cluster, excluding the external computer and the network switch, is instrumented and stored by a digital multimeter (DMM). The evaluation result is shown in Figure 9 and conforms to the simulation results. As shown in Figure 9(a), with all nodes turned on, that is, all-on scenario, all requests are responded to within 1 second, while only approximately 92% of requests can be responded to within 1 second for either *mmm* scenario or *mgm* scenario. On the other hand, both *mmm* scenario and *mgm* scenario consume much less energy than all-on scenario, as shown in Figure 9(b). Similar to the simulation results, the curves of *mgm* and *mmm* are also very close to each other in both Figures 9(a) and 9(b).

VOVO strategy has been studied for more than a decade. Many VOVO approaches [33, 35–37], which dynamically size a cluster according to a preset threshold of CPU utilization or resource usage, were developed based on the designs proposed by Chase et al. [1] or Pinheiro et al. [2]. To compare the proposed queueing-based approach with the threshold-based approaches, Pinheiro's approach [2] is simulated and denoted as *vovo* scenario. In *vovo*, the service demand is smoothed and estimated using the cumulative moving average. *vovo* periodically activates one more node of the cluster when the estimated utilization rate exceeds a predefined threshold and deactivates one node otherwise. The World Cup trace is also used in the simulation of *vovo*. Since *vovo* uses the threshold of CPU utilization rate instead of the response time as a controlling factor, 3 different threshold values, which are 0.7, 0.8, and 0.9, are simulated to get a comparable result.

The simulation results of *vovo* are evaluated with the metric proposed in Section 3.4 and compared with all-on and *mmm*, as shown in Table 3. From this comparison, the threshold of the CPU utilization rate has to be less than 0.8 for *vovo* to get a comparable result with *mmm*. Although *vovo* outperforms *mmm* with the threshold set at 0.7, it requires more nodes and therefore consumes more energy than *mmm*. In order to get a reasonable threshold value for *vovo*, it may be necessary to go through several runs of simulation or other lengthy procedures. On the other hand, the proposed approach minimally requires only the anticipated arrival rate



(a) Response times (b) Energy usage

Figure 9: Real system evaluation.

Table 2: Node profile of the test cluster.

Model	Quantity	Processor	Disk
Acer Veriton N260G	33	1.66 GHz Intel Atom N280	320 GB SATA2 Hitachi HTS54503
Memory	OS	Idle Power	Peak Power
1 GB	Linux 2.6	20.83 W	26.33 W

Table 3: Performance comparison of all-on, *mmm*, and *vovo*.

Log	x	S_x	A_x	E_x	S_x^+	A_x^+	E_x^+	η_x
World Cup	All-on	1.000	1.000	33	1.000	1.000	1.000	1.000
	mmm	0.817	0.993	2.389	1.224	1.007	0.072	0.768
	vovo(0.9)	0.662	0.842	2.162	1.509	1.187	0.066	0.921
	vovo(0.8)	0.812	0.927	2.382	1.231	1.078	0.072	0.794
	vovo(0.7)	0.904	0.970	2.661	1.105	1.030	0.081	0.739

λ, the service rate μ, and the desired response time r to approximate the required number of servers m, that is, (22).

5. Conclusion

This paper proposes two queueing-based sizing methods to periodically adjust the number of servers in a cluster. The proposed method aims at achieving a fair energy-delay performance trade-off of server clusters. The proposed approximation formulas, that is, (22) and (29), are simple closed-form expressions, which may be implemented in a network switch for real-time processing.

From the simulation results, the schemes with the proposed approximation formulas reduce considerable amount of energy consumption while maintaining comparable service performance for gentle service time fluctuations. However, the proposed methods tend to underestimate the number of required servers for service processes with high variability, that is, tail index between 0.3 and 1.3. Similar observation has also been documented in [5].

The relative measurements of *mmm* and *mgm* are almost undifferentiated, except that *mgm* is very slightly better than *mmm* for service processes with high variability. Although Internet workload characterization has found that the probability of service times is a heavy-tailed distribution, periodically resizing the cluster is possible to alleviate the situation of long jobs blocking short jobs in the waiting queue. Because once a deactivation command is issued to a busy node, the node becomes a pending-off node that has to complete the unfinished job before turning itself off. If a long job is handled by this pending-off node, the queued jobs can be quickly assigned to other newly activated nodes of the next period without waiting for the finish of that long job. Therefore, sizing the cluster based on $M/M/m$ model or $M/G/m$ model makes little difference. Based on the simulation results, the simpler $M/M/m$ model may be adequate and preferable for sizing clusters adopting queueing-based VOVO schemes.

Server clusters are widely adopted in cloud data centers [28]. In order to support various kinds of services including user-end applications and back-end activities, heterogeneity becomes common in multifunctional cloud data centers. It is popular that a data center has different group of servers with different computation capacities. Since the basic computing units that are grouped for specific function usually consist of the same type of machines, the proposed approach is built based on the assumption of homogeneous nodes. Therefore, the proposed approach is particularly pertinent for the computing units forming the underling base of cloud data centers. Nevertheless, extending this work to the heterogeneous environments is an immediate future work of this study. The multitier system is an obvious case of server heterogeneity and is widely adopted in many enterprise systems. There are many approaches which have been proposed to address the applicability of queueing models on multitier systems, such as Multitier Internet Applications [25], Heterogeneous Multitier Web Clusters [38], Layered Queueing Networks (LQN) [39, 40], and Power-Saving Server Farms [41]. The job dispatching [42, 43] and scheduling [44, 45] also arise as important issues in a heterogeneous environment. Considering these related developments and integrating the proposed approach with the existing work may be a practical way to extend this study to a heterogeneous environment.

Symbols and Definitions

λ: The job arrival rate of a queueing system
μ: The mean service rate of a server in a queueing system
$1/\mu$: The mean service time of a server in a queueing system
σ: The standard deviation of the service times in a queueing system
m: The number of servers in a queueing system
ρ: The traffic intensity, $\rho = \lambda/m\mu$
j: A system state, which is the same as the number of jobs in the system
p_j: The probability of a state j
C: The coefficient of variation of service times in a queueing system, $C = \sigma\mu$
N: The number of jobs in $M/M/m$ system
N_1: The number of jobs in $M/M/1$ system
M: The number of busy servers in $M/M/m$ system
$E[N]$: The mean value of N
$E[N_1]$: The mean value of N_1
$E[M]$: The mean value of M
R: The response time of a job in $M/M/m$ system
R_1: The response time of a job in $M/M/1$ system
R^+: The response time of a job in $M/G/m$ system
R_1^+: The response time of a job in $M/G/1$ system
W: The waiting time of a job in $M/M/m$ system
W^+: The waiting time of a job in $M/G/m$ system
$E[R]$: The mean value of R
$E[R_1]$: The mean value of R_1
$E[R^+]$: The mean value of R^+
$E[R_1^+]$: The mean value of R_1^+
$E[W]$: The mean value of W
$E[W^+]$: The mean value of W^+.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This study is funded by the Ministry of Science and Technology (Taiwan) under Grant no. NSC 101-2632-E-036-001-MY3 for the project *A Study of Applications and Examinations on the Smart Meter Enabled Electricity Grid*.

References

[1] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, "Managing energy and server resources in hosting centers," *SIGOPS—Operating Systems Review*, vol. 35, pp. 103–116, 2001.

[2] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath, "Load balancing and unbalancing for power and performance in cluster-based systems," in *Proceedings of the Workshop on Compilers and Operating Systems for Low Power (COLP '01)*, vol. 180, pp. 182–195, Barcelona, Spain, 2001.

[3] E. Brockmeyer, H. L. Halstrm, A. K. Erlang, and A. Jensen, *The Life and Works of A.K. Erlang*, Transactions of the Danish Academy of Technical Sciences, Akademiet for de Tekniske Videnskaber, 1948.

[4] R. Guerra, L. Bertini, and J. Leite, "Improving response time and energy efficiency in server clusters," in *Proceedings of the 8th Workshop de Tempo*, p. 8, Curitiba, Brazil, May 2006.

[5] D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: eliminating server idle power," *ACM SIGPLAN Notices*, vol. 44, no. 3, pp. 205–216, 2009.

[6] X. Zheng and Y. Cai, "Markov model based power management in server clusters," in *Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications and International Conference on Cyber, Physical and Social Computing (CPSCom '10)*, pp. 96–102, Washington, DC, USA, 2010.

[7] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges," in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '10)*, pp. 6–17, CSREA Press, 2010.

[8] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, "Optimality analysis of energy-performance trade-off for server farm management," *Performance Evaluation*, vol. 67, no. 11, pp. 1155–1171, 2010.

[9] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M. A. Kozuch, "Autoscale: dynamic, robust capacity management for multi-tier data centers," *ACM Transactions on Computer Systems*, vol. 30, no. 4, article 14, 2012.

[10] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, 1991.

[11] M. Harchol-Balter and A. B. Downey, "Exploiting process lifetime distributions for dynamic load balancing," *ACM Transactions on Computer Systems*, vol. 15, no. 3, pp. 253–285, 1997.

[12] M. E. Crovella, M. S. Taqqu, and A. Bestavros, "Heavy-tailed probability distributions in the world wide web," in *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, R. J. Adler, R. E. Feldman, and M. S. Taqqu, Eds., pp. 3–25, Birkhäuser, Boston, Mass, USA, 1998.

[13] A. Williams, M. Arlitt, C. Williamson, and K. Barker, "Web workload characterization: ten years later," in *Web Content Delivery*, X. Tang, J. Xu, and S. Chanson, Eds., vol. 2 of *Web Information Systems Engineering and Internet Technologies Book Series*, pp. 3–21, Springer, New York, NY, USA, 2005.

[14] D. Ersoz, M. S. Yousif, and C. R. Das, "Characterizing network traffic in a cluster-based, multi-tier data center," in *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS '07)*, p. 59, IEEE, Toronto, Canada, June 2007.

[15] S. Mirtchev and R. Goleva, "Discrete time single server queueing model whit a multimodal packet size distribution," in *Proceedings of the Conjoint Seminar on Modeling and Control of Information Processes*, T. Atanasova, Ed., pp. 83–101, Sofia, Bulgaria, 2009.

[16] M. J. Fischer, D. M. B. Masi, D. Gross, and J. F. Shortle, "One-parameter pareto, two-parameter pareto, three-parameter pareto: is there a modeling difference?" *Alcatel Telecommunications Review*, pp. 79–92, 2005.

[17] A. Gandhi and M. Harchol-Balter, "How data center size impacts the effectiveness of dynamic power management," in *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton '11)*, pp. 1164–1169, September 2011.

[18] D. Meisner, B. T. Gold, and T. F. Wenisch, "The powernap server architecture," *ACM Transactions on Computer Systems*, vol. 29, no. 1, article 3, 2011.

[19] H. Goudarzi, M. Ghasemazar, and M. Pedram, "SLA-based optimization of power and migration cost in cloud computing," in *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid '12)*, pp. 172–179, May 2012.

[20] Z. Liu, Y. Chen, C. Bash et al., "Renewable and cooling aware workload management for sustainable data centers," in *Proceedings of the 12th ACM SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*, pp. 175–186, June 2012.

[21] A. Gandhi, S. Doroudi, M. Harchol-Balter, and A. Scheller-Wolf, "Exact analysis of the M/M/k/setup class of Markov chains via recursive renewal reward," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, pp. 153–166, 2013.

[22] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing server energy and operational costs in hosting centers," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, pp. 303–314, 2005.

[23] D. Meisner, C. M. Sadler, L. A. Barroso, W. Weber, and T. F. Wenisch, "Power management of online data-intensive services," in *Proceeding of the 38th Annual International Symposium on Computer Architecture*, pp. 319–330, San Jose, Calif, USA, June 2011.

[24] Y. Zhang, Y. Wang, and X. Wang, "Electricity bill capping for cloud-scale data centers that impact the power markets," in *Proceedings of the 41st International Conference on Parallel Processing (ICPP '12)*, pp. 440–449, September 2012.

[25] B. Urgaonkar, P. Shenoy, A. Chandra, and P. Goyal, "Dynamic provisioning of multi-tier internet applications," in *Proceedings of the 2nd International Conference on Autonomic Computing (ICAC '05)*, pp. 217–228, June 2005.

[26] V. Gupta, M. Harchol-Balter, J. G. Dai, and B. Zwart, "On the inapproximability of *M/G/K*: why two moments of job size distribution are not enough," *Queueing Systems*, vol. 64, no. 1, pp. 5–48, 2010.

[27] D. Meisner and T. F. Wenisch, "Stochastic queuing simulation for data center workloads," in *Proceedings of the Exascale Evaluation and Research Techniques Workshop*, p. 9, March 2010.

[28] X. Liao, L. Hu, and H. Jin, "Energy optimization schemes in cluster with virtual machines," *Cluster Computing*, vol. 13, no. 2, pp. 113–126, 2010.

[29] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, Wiley-Interscience, 2nd edition, 2001.

[30] M. Arlitt and T. Jin, "Workload characterization study of the 1998 world cup web site," *IEEE Network*, vol. 14, no. 3, pp. 30–37, 2000.

[31] Z. Tari, A. K. A. Phan, M. Jayasinghe, and V. G. Abhaya, *On the Performance of Web Services*, Springer, 2011.

[32] H. Gupta, A. Mahanti, and V. J. Ribeiro, "Revisiting coexistence of poissonity and self-similarity in internet traffic," in *Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS '09)*, pp. 1–10, London , UK, September 2009.

[33] E. N. Elnozahy, M. Kistler, and R. Rajamony, "Energy-Efficient Server Clusters," in *Power-Aware Computer Systems*, B. Falsafi and T. Vijaykumar, Eds., vol. 2325 of *Lecture Notes in Computer Science*, pp. 179–197, Springer, Berlin, Germany, 2003.

[34] J. Nielsen, *Usability Engineering*, Morgan Kaufmann Publishers, 1993.

[35] W. Chen, F. Jiang, W. Zheng, and P. Zhang, "A dynamic energy conservation scheme for clusters in computing centers," in *Embedded Software and Systems*, vol. 3820 of *Lecture Notes in Computer Science*, pp. 244–255, Springer, Berlin, Germany, 2005.

[36] X. Zheng and Y. Cai, "Optimal server provisioning and frequency adjustment in server clusters," in *Proceedings of the 39th International Conference on Parallel Processing Workshops (ICPPW '10)*, pp. 504–511, IEEE, San Diego, Calif, USA, September 2010.

[37] W. Wei, L. Junzhou, S. Aibo, and D. Fang, "Energy-aware dynamic server provisioning and frequency adjustment in multi-tier data centers," *Journal of Internet Technology*, vol. 14, no. 4, pp. 609–618, 2013.

[38] P. Wang, Y. Qi, X. Liu, Y. Chen, and X. Zhong, "Power management in heterogeneous multi-tier web clusters," in *Proceedings of the 39th International Conference on Parallel Processing (ICPP '10)*, pp. 385–394, IEEE, San Diego, Calif, USA, September 2010.

[39] G. Franks, P. Maly, M. Woodside, D. C. Petriu, and A. Hubbard, "Layered queueing network solver and simulator user manual," Tech. Rep., Department of Systems and Computer Engineering, Carleton University, 2005.

[40] Y. Shoaib and O. Das, "Web application performance modeling using layered queueing networks," *Electronic Notes in Theoretical Computer Science*, vol. 275, no. 1, pp. 123–142, 2011.

[41] S. Wang, W. Munawar, X. Liu, and J.-J. Chen, "Power-saving design in server farms for multi-tier applications under

response time constraint," in *Proceedings of the 2nd International Conference on Smart Grids and Green IT Systems (SMARTGREENS '13)*, pp. 137–148, May 2013.

[42] V. Gupta, *Stochastic models and analysis for resource management in server farms [Ph.D. thesis]*, Intel Corporation, 2011.

[43] C.-J. Tang, M.-R. Dai, C.-C. Chuang, Y.-S. Chiu, and W. S. Lin, "A load control method for small data centers participating in demand response programs," *Future Generation Computer Systems*, vol. 32, no. 1, pp. 232–245, 2014.

[44] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, "An analytical model for multi-tier internet services and its applications," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 291–302, 2005.

[45] M. Mazzucco and D. Dyachuk, "Balancing electricity bill and performance in server farms with setup costs," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 415–426, 2012.