

Energy-Efficient Job-Assignment Policy with Asymptotically Guaranteed Performance Deviation

Jing Fu*, *Member, IEEE*, Bill Moran†, *Member, IEEE*

*School of Mathematics and Statistics, the University of Melbourne, VIC3010, Australia

†Department of Electrical and Electronic Engineering, the University of Melbourne, VIC3010, Australia

Email: jing.fu@unimelb.edu.au; wmoran@unimelb.edu.au

Abstract—We study a job-assignment problem in a large-scale server farm system with geographically deployed servers as abstracted computer components (e.g., storage, network links, and processors) that are potentially diverse. We aim to maximize the energy efficiency of the entire system by effectively controlling carried load on networked servers. A scalable, near-optimal job-assignment policy is proposed. The optimality is gauged as, roughly speaking, energy cost per job. Our key result is an upper bound on the deviation between the proposed policy and the asymptotically optimal energy efficiency, when job sizes are exponentially distributed and blocking probabilities are positive. Relying on Whittle relaxation and the asymptotic optimality theorem of Weber and Weiss, this bound is shown to decrease exponentially as the number of servers and the arrival rates of jobs increase arbitrarily and in proportion. In consequence, the proposed policy is asymptotically optimal and, more importantly, approaches asymptotic optimality quickly (exponentially). This suggests that the proposed policy is close to optimal even for relatively small systems (and indeed any larger systems), and this is consistent with the results of our simulations. Simulations indicate that the policy is effective, and robust to variations in job-size distributions.

Index Terms—server farm, energy efficiency, restless multi-armed bandit problem.

I. INTRODUCTION

GLOBAL Internet traffic is rapidly increasing, driving a parallel growth in the data-center industry; over 500 thousand data centers have been launched worldwide [1]. According to an estimate in 2013, about 91 billion kWh of electricity were consumed by U.S. data centers during that year, and the annual consumption has been predicted to reach \$13 billion with nearly 100 million metric tons of carbon pollution, potentially, by 2020 [2]. Servers are considered to be the major contributor to the electrical consumption of data centers [3]. We study the dispatching policies for incoming jobs in a server farm consisting of deployed servers/computer components, aiming to maximize energy efficiency.

Energy-efficient scheduling policies for local server farms are well-studied. For instance, speed scaling technique can reduce energy consumption by decreasing server speed(s) [4], [5]; energy-efficient servers enable dynamic right-sizing of server farms by powering servers on/off/into power-conservative modes according to offered traffic load [6], [7]; and decision making policies according to queue sizes of servers have been considered by changing server working modes [8], [9]. The development of distributed cloud com-

puting platforms has stimulated research on geographically deployed energy-efficient server farms [10], [11].

Server farm vendors deploy a variety of computer components, such as CPUs and disks, to meet various types of inquiries from Internet users, and different generations of these components are present simultaneously because of partial upgrading of old components and purchasing new ones over time [12]. A diversity of physical computing/storage components are available for use in cloud computing platforms, and are abstracted (virtualized) as resources with varying attributes [13]. All of these have resulted in heterogeneity as an important feature in attempting to undertake research on server farms, whereas [14], [15] studied only identical servers.

Regardless of the complexity caused by heterogeneity of server farms, large modern server farms with hundreds of thousands of computer components (abstracted servers) require all scheduling policies to be scalable.

Existing job-assignment policies have been discussed in server farms that have negligible energy consumption on idle servers [4], [8], [16], [17]. Power consumption on idle servers is normally significant in real situations [18], and we so regard it in this paper.

On the other hand, job-assignment policies for network resource allocation problems, such as [19], [20], applicable for practical scenarios with heterogeneous servers and jobs, were studied as static optimizations. Profits to be gained through dynamic release and reuse of resources were ignored. Here we use methods of stochastic optimization that capture dynamic properties of a system.

To maximize the energy efficiency, defined as the ratio of the long-run average throughput to the long-run average power consumption, in a stochastic system with heterogeneous servers, an asymptotically optimal job-assignment policy was proposed in [21]. This optimal policy approaches the optimal solution as the numbers of servers in different server groups tend to infinity proportionately. Nonetheless, the asymptotic optimality is restricted in two aspects: a) although modern server farms are normally large enough to be close to the asymptotic regime, the critical value, above which the numbers of servers are “sufficiently large”, remains unclear, and b) it was assumed that any server in the server farm can serve any arriving job if it has a vacant slot in its buffer. This constraint is not appropriate for geographically separated or functionally varied computer components (abstracted servers). A detailed survey of other published work is provided in Section II.

We aim to maximize the energy efficiency and study the deviation between a newly proposed, scalable policy and the true optimal solution; particularly studying the relationship between this deviation and the number of servers in the system. Also, we extend the idea of [21] to a more general, realistic system in which the available servers for a given job are, to some extent, job-dependent. This extension significantly complicates the entire system and enables consideration of the effects on the problem of the geographical locations and service features (e.g., CPU or memory) of computer components. This was not captured in [21]. We refer to the abstracted servers that are potentially able to serve a job as the *available servers* of this job.

The primary contribution of this paper is a sharpening of the asymptotic optimality results in a heterogeneous server farm, discussed in a special case in [21]. Specifically, we prove that, when the job sizes are exponentially distributed and the blocking probabilities of jobs are always positive, there is a hard upper bound on the deviation between a simple, scalable policy and the optimized energy efficiency in the asymptotic regime; this upper bound *diminishes exponentially* as the number of servers in server groups and the arrival rates of jobs tend to infinity proportionately. In other words, the scalable policy approaches asymptotic optimality quickly (exponentially) as the size of the optimization problem increases. We refer to this upper bound as the *deviation bound*, and the policy as *Priorities accounting for Available Servers* (PAS), as it is a priority-style policy and applicable for a system with different sets of available servers.

Our secondary contributions are twofold:

- We consider a large-scale system, potentially containing several geographically distributed server farms, and regard this hybrid system as an abstracted server farm model. We propose a scalable PAS policy in this server farm with heterogeneous servers and jobs, where energy consumption and service rates of servers can be arbitrarily related. As mentioned in our primary contribution, the PAS policy is proved to rapidly (exponentially) approach asymptotic optimality as the scale of the server farm increases. To the best of our knowledge, no existing work has studied the energy efficiency of such a realistically scaled heterogeneous server farm, nor does any existing work propose a scalable scheduling policy with proven guaranteed performance.
- By numerical simulations, we demonstrate that PAS is nearly optimal even for relatively small server farms. Together with the rapidly decreasing deviation bound, when job sizes are exponentially distributed and the blocking probabilities of jobs are always positive, it is likely to be near-optimal in all larger server farms and proved to approach optimality as the server farm sizes tend to infinity. In particular, the deviation bound of PAS is demonstrated to decrease with increasing server farm size, consistent with our theoretical results, and to be less than 3% in all our experiments involving only 100 servers (computer components). Also, we numerically demonstrate that the PAS policy is relatively insensitive to the specific job-size distribution by comparing its energy

efficiency with different job-size distributions.

The paper is organized as follows: Section II: discussion of related work on job assignment policies; Section III: description of the server farm model; Section IV: definition of the stochastic optimization problem; Section V: description of the PAS policy; Section VI: proof of the deviation bound; Section VII: numerical results; Section VIII: conclusions.

II. OTHER RELATED WORK

Asymptotically optimal job-assignment policies applicable to a parallel queueing model with infinite buffer size on each server were studied in [22], [23]. In [24], [25], policies were proposed for server farms without capacity constraints (with infinite buffer size), aimed at minimization of average delay. A detailed survey of asymptotically optimal job-assignment policies was given in [21].

In particular, the asymptotic results presented in this paper are obtained by implementing the Whittle relaxation technique, which was originally designated for the *restless multi-armed bandit problem* (RMABP) [26]. The optimization of a general RMABP was proved to be PSPACE-hard [27]; nonetheless, in [28], a scalable policy, proposed through the Whittle relaxation technique and referred to as the (*Whittle*) *index policy* in [26], was proved to be asymptotically optimal under conditions that require a global attractor of a stochastic process associated with the RMABP, and *Whittle indexability* [26]. There is not a necessary implication between one of these conditions and asymptotic optimality. In [29], [30], Niño-Mora proposed and analyzed the *Partial Conservation Law (PCL) indexability* for the performance of scheduling problems, such as the (restless) multi-armed bandit problem. This implies (and is stronger than) the Whittle indexability. A More detailed survey about indexability can be found in [31].

Other publications that discuss management of jobs in server farms by distributing offered traffic were published recently [5], [10]. Lenhart *et al.* [5] provided an experimental analysis on energy-efficient web servers with an assumption of a cubic relationship between server power consumption and traffic load. The number of servers (nodes) tested in [5] is very small compared to a real system in modern data centers. Lin *et al.* [10] analyzed energy efficiency performance in communication networks in a server farm model. They assumed power consumption is negligible on idle servers (multiplexing/aggregating nodes in network). Recall that, here, we allow the possibility of positive power consumption on idle.

An auction based mechanism was studied in a server farm model with heterogeneous servers (resources) and job types (users) in [11]. The authors provided a worst-case ratio (competitive ratio) of the revenue (social welfare) under their proposed policies relative to the optimal solution, and showed it to be linearly increasing in the time horizon. Also, in [11], the length of each job is assumed known before assigning it. Here, we consider the more realistic situation that the job length remains unknown until it is completed.

Stochastic job-assignment techniques were studied in [32], where a linearly increasing relationship of reward/cost rates to traffic loads of servers is assumed. They proposed a reasonable and scalable policy with a given parameter V and proved the

deviation between this policy and the optimal solution to be $O(1/V)$, where V is a parameter related to the Lyapunov optimization technique. For the version of energy-efficient server farms considered here, the energy consumption rate of computer components is generally non-linear in its traffic load, so the linearity is not appropriate for modeling energy consumption rates of computer components in Cloud environments. In consequence, the results in [32] are not directly applicable to the energy-efficient server farm.

In [32], the blocking of jobs was allowed when not all servers are fully occupied. Also, a deterministic job lifespan model was assumed and all server were assumed to be able to handle all jobs. We strictly prohibit blocking of jobs when available servers are not fully occupied, to ensure the fairness for all customers. We also consider a diversity of jobs with randomly generated job sizes (remaining unknown until completed) and with job-dependent sets of available servers.

In summary, the non-linearity of power functions and the complexity of our server farm model prevent applicability of existing methods from being direct. Also, there is no published work that provided theoretical bounds that are quickly (exponentially) decreasing in the number of servers between proposed policy and the optimal solution.

Moreover, as mentioned in Section I, powering servers on/off [6], [14], [15] or switching into power conservative modes with additional suspending time [7], [8], [10], [32] enables dynamic variation of the size/number of working of working servers in a server farm. In [6], [15], such server farms are called *dynamic right-sizing* server farms. In similar vein to [16], [17], [21], for the purposes of this paper, a fixed number of working servers is postulated in a server farm with no possibility of powering off or state switching, with substantial delay, during the time period under consideration. In practice, this corresponds to periods during which no powering off or state switching, with concomitant substantial delays, takes place. In this context, the job assignment policies discussed here can be combined with the right-sizing techniques, as appropriate. Note that frequent powering off/on or state switches increases wear and tear of hardware and ensuing requirements for costly replacement and maintenance [33].

III. MODEL

We classify incoming jobs into J job types labeled by an integer j ($j = 1, \dots, J$), each of which has an arrival rate $\lambda_j > 0$ indicating the average number of arrivals per unit time, following a Poisson process as previous studies in Cloud environments [34], [35]. If groups/type of Internet/network customers decide to send requests independently and identically during a given time period and the number of such customers is sufficiently large for the corresponding dynamic process to become stationary, then it is reasonable to model the arrival process of customer requests for this type as a Poisson process, although the arrival rates may vary from one time period to another, which is consistent with observations of real-world tracelogs [36], [37].

These jobs will be undertaken by servers or blocked. We classify the servers into different server groups according to their functional features and profiles. Define the set of server

groups as $\mathcal{K} := \{1, 2, \dots, K\}$. We assume that there are in total $K \geq 2$ server groups and $R_k \geq 1$ servers in group k . Each job type j is only able to be served by a server from a subset $\emptyset \neq \mathcal{K}_j \subset \mathcal{K}$ of server groups. We say that $k \in \mathcal{K}_j$ is an *available server group* for job type j and a server of group k is an available server for a job of type j . A server's availability of serving different jobs can be affected by its functional features, jobs' preferences, and geographical distances from the jobs.

The *sizes* of jobs of the same type are independent identically distributed (i.i.d.) with average job size normalized to be one (bit). We assume, for convenience, i.i.d. job sizes across all job types for the theoretical development, but we provide numerical results in Section VII to indicate the robustness of our results when when job-size distributions vary across job types.

A server of group k serves its jobs at a total and peak rate of μ_k using the processor sharing (PS) service discipline. When the server is not idle, the service rate received by each job is then μ_k divided by the number of jobs in the server buffer. The service rate of each server in group k is supported by consuming non-negligible amount of energy, even in its idle mode. Similarly, we assume that each server operates in a power-consuming mode with its peak energy consumption rate when there are jobs accommodated. We refer to this peak energy consumption rate as the energy consumption rate of a busy server, denoted by ε_k for group $k = 1, 2, \dots, K$. When the server is idle, it automatically changes to a power-conservative mode and consumes ε_k^0 energy per unit time. Evidently, $\varepsilon_k > \varepsilon_k^0 \geq 0$. The service rates and energy consumption rates of busy/idle servers are intrinsic parameters determined by the server hardware features and profiles, and the relationship between them can be arbitrary in this paper.

As in [6], [8], [16], [17], [21], the busy/idle operating rule is more appropriate to machines working in two-power modes, such as Oracle Sun Fire X2270 M2, Cisco UCS C210, Cisco MXE 3500 and Cisco UCS 5108. Also, the IEEE 802.11 standards define exactly two power saving modes for network components in energy-conservative communications.

Moreover, we study a system with finite service capacity on each server, referred to as its *buffer size*. It provides a finite bound on the number of jobs being simultaneously served by this server. Let $B_k \geq 1$ represent the buffer size of a server in group k .

A *policy* ϕ is a mechanism to assign an arriving job of type j to a server of group $k \in \mathcal{K}_j$ with at least one vacant slot in its buffer. A fairness criterion requires equal treatment of the different job types. In particular, rejection of jobs sent by different users is not permitted if there are vacant slots on available servers. If there is no such server (all available buffers are fully occupied), the arriving job is lost.

Consider a ratio of total arrival rate to the total service rate, i.e., $\rho := \sum_{j \in \mathcal{J}} \lambda_j / \sum_{k \in \mathcal{K}} \mu_k$, the *normalized offered traffic* (see [21]). For a specific job type j , the *normalized offered traffic of type j* , $\rho_j := \lambda_j / \sum_{k \in \mathcal{K}_j} \mu_k$.

We assume the existence of long-run averages of throughput and power consumption, and refer to them as the *job throughput* and the *power consumption* of the system, respectively. Precise definitions are given in Section IV. We define \mathcal{L}^ϕ and

\mathcal{E}^ϕ to be the job throughput and power consumption of the system under policy ϕ , respectively. The *energy efficiency of policy ϕ* , $\mathcal{L}^\phi/\mathcal{E}^\phi$, is the objective of our problem for energy-efficient server farms. The value of the energy efficiency indicates the average job throughput achieved by consuming one unit energy. Since we do not permit rejection of jobs when there are vacancies on available servers, the objective encapsulates a trade-off between performance and energy consumption.

IV. STOCHASTIC OPTIMIZATION PROBLEM

We study a stochastic system, where the dynamically released service capacities of physical components can be reused. To capture the stochastic features, we start by introducing the stochastic state of the entire system: a server farm with tens of thousands of heterogeneous servers.

Define the *state of an individual server* as the number of jobs currently on the server. The set of all possible states of a server of group $k \in \mathcal{K}$, is denoted by $\mathcal{B}_k = \{0, 1, \dots, B_k\}$, where $B_k \geq 1$. As in [21], states $0, 1, \dots, B_k - 1$ are called *controllable*, and the state B_k is *uncontrollable*, since all new jobs will be rejected by a server in state B_k . Denote the set of all controllable states of server group k by $\mathcal{C}_k = \{0, 1, \dots, B_k - 1\}$ and that of the uncontrollable states by $\mathcal{U}_k = \{B_k\}$.

Servers of the same group have potentially different states in the stochastic process. Define the set of servers in group k as \mathcal{R}_k . The set of all servers is $\mathcal{S} := \bigcup_{k \in \mathcal{K}} \mathcal{R}_k$, the set of servers available for jobs of type $j \in \mathcal{J}$ is $\mathcal{S}_j := \bigcup_{k \in \mathcal{K}_j} \mathcal{R}_k$, and the state space of the entire system is $\mathcal{B} := \prod_{k \in \mathcal{K}} (\mathcal{B}_k)^{R_k}$. The size of the state space thus increases exponentially in the number of servers in the server farm, i.e., $|\mathcal{B}|$, which, in itself, is normally very large in modern real server farms.

Decisions driven by a stationary policy ϕ applied to job arrivals rely on the state of the system just before an arrival occurs and the information known in this state, such as average rates of transitioning to other states. For a policy ϕ , $s \in \mathcal{S}$, $j \in \mathcal{J}$, we define the action variable $a_{j,s}^\phi(\mathbf{n}) \in \{0, 1\}$, $\mathbf{n} \in \mathcal{B}$, to indicate the decision under policy ϕ for an arriving job of type j on server s in state \mathbf{n} : server s accepts an arriving job of type j if $a_{j,s}^\phi(\mathbf{n}) = 1$; and does not accept any job otherwise. In this context, for each job of type j , there is at most one server s with $a_{j,s}^\phi(\mathbf{n}) = 1$ among all available servers.

In addition, we define, for $\mathbf{n} \in \mathcal{B}$, $j \in \mathcal{J}$, $s \in \mathcal{S}$,

- $a_{j,s}^\phi(\mathbf{n}) \equiv 0$ if $s \notin \mathcal{S}_j$ to prevent a server that is unavailable for job type j ;
- $a_{j,s}^\phi(\mathbf{n}) \equiv 0$ if $n_s = B_k$, $s \in \mathcal{R}_k$, $k \in \mathcal{K}_j$, to prevent a server from accepting a new job when it is fully occupied.

Then, the action space, as the discrete set of possible values of the action variables, is

$$\mathcal{A} := \prod_{j \in \mathcal{J}} \{0, 1\}^{\sum_{k \in \mathcal{K}_j} R_k}, \quad (1)$$

where we recall that \mathcal{K}_j is the set of available server groups for job type j and R_k is the number of servers in group k . With large number of servers R_k , multiple job types (i.e., $J > 1$) significantly enlarge the number of action variables in parallel with the size of the action space.

Let $N_s^\phi(t) \in \mathcal{B}_k$, $s \in \mathcal{R}_k$, $k \in \mathcal{K}$, represent the state of server s at time t under policy ϕ , and $\mathbf{N}^\phi(t) = (N_s^\phi(t) : s \in \mathcal{S}) \in \mathcal{B}$. For simplicity, we always consider an empty system at time 0, $\mathbf{N}^\phi(0) = \mathbf{0}$.

It will be convenient to consider mappings $\mathbf{f} = (f_1, f_2, \dots, f_K)$, where $f_k : \mathcal{B}_k \rightarrow \mathbb{R}$. We refer to such a vector of mappings $\mathbf{f} \in \prod_{k \in \mathcal{K}} \mathbb{R}^{\mathcal{B}_k}$ as the vector of *reward rate functions*. Then, for a given $\mathbf{f} \in \prod_{k \in \mathcal{K}} \mathbb{R}^{\mathcal{B}_k}$, for some k , we define the *long-run average performance* of the system under policy ϕ to be

$$\Gamma^\phi(\mathbf{f}) = \lim_{t \rightarrow +\infty} \frac{1}{t} \mathbb{E} \left[\int_0^t \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{R}_k} f_k(N_s^\phi(u)) du \right], \quad (2)$$

where we assume the existence of such limit.

Specifically, we define $f_k^\mu(n_k)$ and $f_k^\varepsilon(n_k)$, $n_k \in \mathcal{B}_k$, $k \in \mathcal{K}$, as the service rate and energy consumption rate of a server in group k in state n_k , respectively, and consider them as the reward rate functions; that is, for $k \in \mathcal{K}$, f_k^μ and f_k^ε are mappings: $\mathcal{B}_k \rightarrow \mathbb{R}$. As defined in Section III, $f_k^\mu(n_k) = \mu_k$, $f_k^\varepsilon(n_k) = \varepsilon_k$ for $n_k > 0$, $f_k^\mu(0) = 0$ and $f_k^\varepsilon(0) = \varepsilon_k^0$, where $\mu_k > 0$, $\varepsilon_k > \varepsilon_k^0 \geq 0$, $k \in \mathcal{K}$. For the vectors $\mathbf{f}^\mu = (f_1^\mu, f_2^\mu, \dots, f_K^\mu)$ and $\mathbf{f}^\varepsilon = (f_1^\varepsilon, f_2^\varepsilon, \dots, f_K^\varepsilon)$, the job throughput of the entire system is, then, $\Gamma^\phi(\mathbf{f}^\mu)$, and the power consumption of the system is $\Gamma^\phi(\mathbf{f}^\varepsilon)$. Recall that our objective is to maximize the energy efficiency of the entire system; that is, $\mathcal{L}^\phi/\mathcal{E}^\phi = \Gamma^\phi(\mathbf{f}^\mu)/\Gamma^\phi(\mathbf{f}^\varepsilon)$.

To complete necessary constraints on the action variables of our optimization problem, there remain definitions to accommodate behavior that *blocks* an arriving job. We define a virtual server group 0 with server number $R_0 = 1$ and server set $\mathcal{R}_0 = \{1\}$, which receives blocked jobs. Any server of group 0 has only one state with zero transition rate all the time: that is, it does not generate any reward or cost to the entire system. We define \mathcal{B}_0 as the state space of a server of group 0 where $|\mathcal{B}_0| = 1$. Also, the set of controllable states for 0 is set to be $\mathcal{C}_0 = \mathcal{B}_0$ and the one for uncontrollable states is $\mathcal{U}_0 = \emptyset$.

We extend the original definition of a policy ϕ determined by actions $a_{j,s}^\phi(\mathbf{n})$ ($\mathbf{n} \in \mathcal{B}$, $s \in \mathcal{S}$, $j \in \mathcal{J}$), to that determined by actions $(a_{j,s}^\phi(\mathbf{n}) : \mathbf{n} \in \mathcal{B}$, $s \in \mathcal{S}$, $j \in \mathcal{J}$; $a_j^\phi(\mathbf{n}))$, where $a_j^\phi(\mathbf{n}) \in \{0, 1\}$ represents the action variable for the only server of virtual group 0 for job type j . We slightly abuse the notation and still use ϕ to denote such a policy.

Let $k(s)$ be the label of the server group satisfying $s \in \mathcal{R}_{k(s)}$, and $I(x)$ be the Heaviside function: for $x \in \mathbb{R}$, $I(x) = 1$ if $x > 0$; and 0 otherwise. Define $\Theta(x) = xI(x)$, $x \in \mathbb{R}$. Our problem is then encapsulated by

$$\max_{\phi} \Gamma^\phi(\mathbf{f}^\mu) / \Gamma^\phi(\mathbf{f}^\varepsilon) \quad (3)$$

with policy ϕ subject to

$$\sum_{s \in \mathcal{S}_j} a_{j,s}^\phi(\mathbf{N}^\phi(t)) + \Theta(\bar{a}_j^\phi(\mathbf{N}^\phi(t))) = 1, \forall j \in \mathcal{J}, t \geq 0, \quad (4)$$

$$\bar{a}_j^\phi(\mathbf{N}^\phi(t)) + \sum_{s \in \mathcal{S}_j} I(B_{k(s)} - N_s^\phi(t)) \leq 1, \forall j \in \mathcal{J}, t \geq 0, \quad (5)$$

by introducing variables $\bar{a}_j^\phi(\mathbf{n}) \in \mathbb{R}$ and setting $a_j^\phi(\mathbf{n}) = \Theta(\bar{a}_j^\phi(\mathbf{n}))$, $\mathbf{n} \in \mathcal{B}$. Define Φ to be the set of all the policies ϕ

satisfying (4) and (5). Constraints in (5) and variables $\bar{a}_j^\phi(\mathbf{n})$ are introduced to guarantee that jobs of type j are blocked if and only if all the available servers are fully occupied:

- if all available servers for job type j are fully occupied at time t , then, because all available servers are in uncontrollable states, $a_{j,s}^\phi(\mathbf{N}^\phi(t)) = 0$ for all $s \in \mathcal{S}_j$ and the constraint in (4) forces $a_j^\phi(\mathbf{N}^\phi(t)) = \Theta(\bar{a}_j^\phi(\mathbf{N}^\phi(t))) = 1$, which does not violate the constraint in (5);
- otherwise, the constraint in (5) forces $\bar{a}_j^\phi(\mathbf{N}^\phi(t)) \leq 0$, which leads to $a_j^\phi(\mathbf{N}^\phi(t)) = \Theta(\bar{a}_j^\phi(\mathbf{N}^\phi(t))) = 0$ and so a newly-arrived job of type j cannot be blocked.

We aim at a largely scaled problem that exhibits inevitably high computational complexity. A special case of our problem is in fact an instance of the Restless Multi-Armed Bandit Problem (RMABP) [26]. The RMABP has been proved to be PSPACE-hard [27] in the general case, so that near-optimal, scalable approximations are the impetus to this paper. Here, we consider a general case involving not only heterogeneous servers, but also heterogeneous jobs, which is much more practical and significantly increases the size of the action space, as described in (1). In particular, different sets of available servers for different jobs enable consideration of geographically distributed servers, jobs' preferences and functional differences among servers.

V. PRIORITY-STYLE POLICY

As mentioned in Section I, in [21], a policy that always prioritizes the most efficient servers was proposed and proved to approach the optimality when the problem size (number of servers) becomes arbitrarily large; in other words, it is asymptotically optimal. However, this policy requires that the set of available servers for each job always include all servers in the server farm, and the asymptotic optimality is not applicable in a realistic system with a large but finite number of servers. It is important to know the detailed relationship between the performance degradation and the problem size.

Recall our objective of maximizing the energy efficiency of the entire server farm, defined as the ratio of the job throughput to the power consumption. The power consumption can be interpreted as the cost used to support corresponding job throughput. In this context, for an idle server in group $k \in \mathcal{K}$, ε_k^0 units of power are consumed in support of a zero service rate; if the server becomes busy, $\varepsilon_k - \varepsilon_k^0$ power is added to produce a service rate μ_k .

In other words, the idle power ε_k^0 is a persistent and uncontrollable cost producing no service rate; while, $\varepsilon_k - \varepsilon_k^0$ is the productive and controllable part of the power that serves jobs at service rate μ_k . We propose a policy that always prioritizes servers producing higher service rates per unit controllable power; namely, the ratio of its service rate to the productive part of its power consumption, $\mu_k/(\varepsilon_k - \varepsilon_k^0)$. The ratio was referred to as the *effective energy efficiency* in [21].

In particular, for an incoming job of type $j \in \mathcal{J}$ with a set of available servers \mathcal{S}_j , the job will be assigned to a server in \mathcal{S}_j with highest effective energy efficiency and with at least one vacancy in its buffer. As indicated earlier, we refer to such a policy as the *Priorities accounting for Available Servers* (PAS). Note that PAS uses a similar idea to that proposed in

Algorithm 1: Updating the indication vector upon an arrival.

Input : The server $s \in \mathcal{S}$ where an arrival occurs at time t .
Function UpdatingUponArrival(s):
 for $j \in \mathcal{J}$ with $s \in \mathcal{S}_j$ **do**
 Remove s from the max heap $\mathcal{H}_j(t)$
 $v_j(t) \leftarrow$ the root node of the updated $\mathcal{H}_j(t)$
 end
return

Algorithm 2: Updating the indication vector upon a departure.

Input : The server $s \in \mathcal{S}$ where a departure occurs at time t .
Function UpdatingUponDeparture(s):
 for $j \in \mathcal{J}$ with $s \in \mathcal{S}_j$ **do**
 Add s in the max heap $\mathcal{H}_j(t)$
 $v_j(t) \leftarrow$ the root node of the updated $\mathcal{H}_j(t)$
 end
return

[21], but is applicable for our problem with different sets of available servers.

For $j \in \mathcal{J}$, let $\mathcal{N}_j^{\{0\}} = \{\mathbf{n} \in \mathcal{B} \mid \forall s \in \mathcal{S}_j, n_s \in \mathcal{U}_{k(s)}\}$ and $\mathcal{N}_j^{\{0,1\}} = \{\mathbf{n} \in \mathcal{B} \mid \exists s \in \mathcal{S}_j, n_s \in \mathcal{C}_{k(s)}\}$. Rigorously, the action variables for PAS are given by

$$a_{j,s}^{\text{PAS}}(\mathbf{n}) = \begin{cases} 1, & \text{if } \mathbf{n} \in \mathcal{N}_j^{\{0,1\}} \text{ and} \\ & s \in \arg \max_{s \in \mathcal{S}_j : n_s \in \mathcal{C}_{k(s)}} \left[\frac{\mu_{k(s)}}{\varepsilon_{k(s)} - \varepsilon_{k(s)}^0} \right], \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

If $\arg \max[\cdot]$ returns a set with more than one element, ties can be broken arbitrarily. We set, without loss of generality, policy PAS to always select the smallest s among the set of value(s) returned by this $\arg \max[\cdot]$.

For clarity, we provide an example of implementing PAS. Maintain a *indication vector* $\mathbf{v}(t) \in \mathcal{S}^{|\mathcal{J}|}$ at time t , where the j th element $v_j(t)$ represents the server used to accommodate a newly-arrive job of type j if it arrives at time t . Note that, $a_{j,s}^{\text{PAS}}(\cdot)$ can be determined by $\mathbf{v}(t)$ by setting $a_{j,s}^{\text{PAS}}(\cdot) = 1$ for $s = v_j(t)$ and 0 for others. For each job type $j \in \mathcal{J}$, maintain a max heap $\mathcal{H}_j(t)$ of servers $s \in \mathcal{S}_j$ with respect to the effective energy efficiency $\mu_{k(s)}/(\varepsilon_{k(s)} - \varepsilon_{k(s)}^0)$. If a server s transitions from $B_{k(s)} - 1$ to $B_{k(s)}$ or from $B_{k(s)}$ to $B_{k(s)} - 1$, we trigger a potential update of the indication vector and the max heaps for all types, as described in Algorithms 1 and 2, respectively. For both algorithms, the worst-case computational complexity is $O(\sum_{j \in \mathcal{J}} \log |\mathcal{S}_j|)$ and the space complexity is $O(\sum_{j \in \mathcal{J}} |\mathcal{S}_j|)$, representing the storage space for the vector $\mathbf{v}(t)$ and heaps $\mathcal{H}_j(t)$ ($j \in \mathcal{J}$).

Note that the PAS policy does not require λ_j to be known nor, indeed, do we assume specific distributions for the inter-arrival/inter-departure times. In other words, PAS is widely applicable and scalable to a server farm with heterogeneous servers and jobs.

VI. DEVIATION ANALYSIS

Following similar ideas in [21], [28], in this section, we obtain an upper bound for the performance deviation between PAS and the optimal solution in asymptotic regime, under certain condition. We refer to this upper bound as the *deviation bound*. The deviation bound diminishes exponentially in the size of the system leading, in particular, to asymptotic optimality of PAS.

Following the idea in [26], servers should be prioritized according to their potential profits, quantized and obtained by relaxing the constraint of the optimization problem. This is referred to as the *Whittle relaxation* technique. Our problem, (3)-(5), is treated similarly.

This technique produces a highly intuitive heuristic scheduling policy, which coincides with PAS under certain conditions. We prove this equivalence in Section VI-A. Based on this equivalence, in Section VI-B, we prove that the deviation bound of PAS's performance is diminishing rapidly (in fact, exponentially) as the problem size increases.

A. Whittle Relaxation

From [16, Theorem 1], if $\Gamma^\phi(\mathbf{f}^\mu) < +\infty$ and $\Gamma^\phi(\mathbf{f}^\varepsilon) < +\infty$ for all $\phi \in \Phi$, and we define,

$$e^* = \max_{\phi \in \Phi} \{ \Gamma^\phi(\mathbf{f}^\mu) / \Gamma^\phi(\mathbf{f}^\varepsilon) \}, \quad (7)$$

then a policy optimizes the problem described by equations (3)-(5) if and only if it optimizes

$$\max_{\phi} \{ \Gamma^\phi(\mathbf{f}^r) \}, \text{ s. t. (4) and (5),} \quad (8)$$

where the vector of reward rate functions $\mathbf{f}^r \in \prod_{k \in \mathcal{K}} \mathbb{R}^{\mathcal{B}_k}$ is defined by

$$f_k^r(n_k) = f_k^\mu(n_k) - e^* f_k^\varepsilon(n_k), \quad (9)$$

for $n_k \in \mathcal{B}_k$, $k \in \mathcal{K}$. Let

$$\alpha_j^\phi = \lim_{t \rightarrow +\infty} \mathbb{E}[a_{j,s}^\phi(\mathbf{N}^\phi(t))] = \lim_{t \rightarrow +\infty} \mathbb{E} \left[\Theta \left(\bar{a}_j^\phi(\mathbf{N}^\phi(t)) \right) \right]$$

and

$$\bar{\alpha}_j^\phi = \lim_{t \rightarrow +\infty} \mathbb{E} \left[\bar{a}_j^\phi(\mathbf{N}^\phi(t)) \right].$$

The Whittle relaxation technique involves randomization of the action variables $a_{j,s}^\phi(\cdot)$ and $\bar{a}_j^\phi(\cdot)$. This relaxation of (3)-(5) produces the following problem:

$$\max_{\phi} \Gamma^\phi(\mathbf{f}^r) \quad (10)$$

subject to

$$\sum_{s \in \mathcal{S}_j} \lim_{t \rightarrow +\infty} \mathbb{E} \left[a_{j,s}^\phi(\mathbf{N}^\phi(t)) \right] + \alpha_j^\phi = 1, \quad \forall j \in \mathcal{J}, \quad (11)$$

$$\bar{\alpha}_j^\phi + \sum_{s \in \mathcal{S}_j} \lim_{t \rightarrow +\infty} \mathbb{E} \left[I(B_{k(s)} - N_s^\phi(t)) \right] \leq 1, \quad \forall j \in \mathcal{J}. \quad (12)$$

The relaxed problem no longer captures the server farm problem realistically, but is useful for theoretical analysis. Define

$$A_j = \sum_{s \in \mathcal{S}_j} \sum_{n \in \mathcal{C}_{k(s)}} \pi_s^*(n), \quad j \in \mathcal{J},$$

where $\pi_s^*(n)$ is the steady-state probability of state n for server s under policy ϕ satisfying $a_{j',s'}^\phi(\mathbf{n}') = 1$ for all $\mathbf{n}' \in \mathcal{B}$, $n'_{s'} \in \mathcal{C}_{k(s')}$, $j' \in \mathcal{J}$ and $s' \in \mathcal{S}_{j'}$. It is clear that, for $j \in \mathcal{J}$,

$$A_j \leq \sum_{s \in \mathcal{S}_j} \lim_{t \rightarrow +\infty} \mathbb{E} \left[I(B_{k(s)} - N_s^\phi(t)) \right].$$

In this context, Equation (12) can be relaxed further to

$$\Theta(\bar{\alpha}_j^\phi) \leq \Theta(1 - A_j), \quad \forall j \in \mathcal{J}. \quad (13)$$

To complete the analysis, we introduce, for the relaxed

problem defined by (10),(11) and (13), a redundant constraint:

$$\lim_{t \rightarrow \infty} \mathbb{E}[a_{j,s}^\phi(\mathbf{N}^\phi(t)) | N_s^\phi = B_{k(s)}] = 0, \quad \begin{cases} \forall j \in \mathcal{J}, \\ \forall s \in \mathcal{S}_j, \end{cases} \quad (14)$$

which forces the action variables for the uncontrollable states to be zero. Constraints (11), (13) and (14) can be combined with the objective function by introducing Lagrangian multipliers ν , γ and η corresponding to (11), (13) and (14), respectively. Let $\tilde{\Phi}$ represent the set of all stationary policies. Define

- 1) $\pi_s^\phi(n)$, $s \in \mathcal{S}$, $n \in \mathcal{B}_{k(s)}$, $\phi \in \tilde{\Phi}$, as the steady state probability of server s in state n under policy ϕ ;
- 2) row vector $\pi_s^\phi = (\pi_s^\phi(n) : n \in \mathcal{B}_{k(s)})$, $s \in \mathcal{S}$, $\phi \in \tilde{\Phi}$;
- 3) column vector $\mathbf{f}_s^r = (f_s^r(n) : n \in \mathcal{B}_{k(s)})$, $s \in \mathcal{S}$;
- 4) column vector $\alpha_{j,s}^\phi = (\alpha_{j,s}^\phi(n) : n \in \mathcal{B}_{k(s)})$ where

$$\alpha_{j,s}^\phi(n) = \lim_{t \rightarrow +\infty} \mathbb{E} \left[a_{j,s}^\phi(\mathbf{N}^\phi(t)) | N_s^\phi(t) = n \right],$$

$$j \in \mathcal{J}, s \in \mathcal{S}_j, \phi \in \tilde{\Phi};$$

- 5) column vector e^n , $n \in \mathbb{N}^+$, of size n with all zero elements except the n th element set to be one.

The Lagrange problem with respect to the primal problem defined by (10), (11), (13) and (14) is then

$$\begin{aligned} \Lambda(\nu, \gamma, \eta) = \\ \max_{\phi} \sum_{s \in \mathcal{S}} \pi_s^\phi \left(\mathbf{f}_s^r - \sum_{j \in \mathcal{J}_s} \nu_j \alpha_{j,s}^\phi - \sum_{j \in \mathcal{J}_s} \eta_{j,s} \alpha_{j,s}^\phi(B_{k(s)}) e^{B_{k(s)}} \right) \\ - \sum_{j \in \mathcal{J}} (\nu_j \alpha_j^\phi + \gamma_j \Theta(\bar{\alpha}_j^\phi)) + \sum_{j \in \mathcal{J}} (\nu_j + \gamma_j \Theta(1 - A_j)) \end{aligned} \quad (15)$$

where $\mathcal{J}_s = \{j \in \mathcal{J} | s \in \mathcal{S}_j\}$, $s \in \mathcal{S}$.

As in [26], given ν , γ and η , the maximization problem at the right hand side of (15) achieves the same maximum as a sum of the maximum values of $|\mathcal{S}| + J$ independent sub-problems: for $s \in \mathcal{S}$,

$$\max_{\phi \in \Phi_s} \pi_s^\phi \left(\mathbf{f}_s^r - \sum_{j \in \mathcal{J}_s} \nu_j \alpha_{j,s}^\phi - \sum_{j \in \mathcal{J}_s} \eta_{j,s} \alpha_{j,s}^\phi(B_{k(s)}) e^{B_{k(s)}} \right) \quad (16)$$

where Φ_s represents the set of stationary policies ϕ determined by action variables $\alpha_{j,s}^\phi(n) \in [0, 1]$, $n \in \mathcal{B}_{k(s)}$, $j \in \mathcal{J}$; and for $j \in \mathcal{J}$,

$$\max_{\phi \in \bar{\Phi}_j} -\nu_j \alpha_j^\phi - \gamma_j \Theta(\bar{\alpha}_j^\phi), \quad (17)$$

where $\bar{\Phi}_j$ is the set of stationary policies ϕ determined by action variables $\bar{\alpha}_j^\phi$ and α_j^ϕ . Remarkably, the dimension of the state space for each of these independent sub-problems is 1.

Condition 1. For all $j \in \mathcal{J}$, $A_j \leq 1$.

We refer to Condition 1 as the *heavy traffic condition*: the blocking probabilities for jobs are *almost* positive all the time.

Proposition 1. When the job sizes are exponentially distributed, if either $J = 1$ or Condition 1 holds true, then there exists a $\nu \in \mathbb{R}$ and a policy $\phi^* \in \Phi$ that maximizes the relaxed problem defined by (10), (11), (13) and (14), satisfying, for $j \in \mathcal{J}$,

$$1) \text{ if } s \in \mathcal{S}_j, n \in \mathcal{C}_{k(s)}, \quad \alpha_{j,s}^{\phi^*}(n) = \begin{cases} 1, & \text{if } \nu < 1 - e^* \frac{\varepsilon_{k(s)} - \varepsilon_{k(s)}^0}{\mu_{k(s)}} \\ 1 \text{ or } 0, & \text{if } \nu = 1 - e^* \frac{\varepsilon_{k(s)} - \varepsilon_{k(s)}^0}{\mu_{k(s)}} \\ 0, & \text{if } \nu > 1 - e^* \frac{\varepsilon_{k(s)} - \varepsilon_{k(s)}^0}{\mu_{k(s)}}; \end{cases} \quad (18)$$

2) and

$$\alpha_j^{\phi^*} = 1 - \sum_{s \in \mathcal{S}_j} \sum_{n \in \mathcal{C}_{k(s)}} \pi_s^{\phi^*}(n) \alpha_{j,s}^{\phi^*}(n). \quad (19)$$

The proof of Proposition 1 is given in Appendix A.

Note that although we can calculate the optimal solution of the relaxed problem, referred to as policy ϕ^* in Proposition 1, such ϕ^* is not applicable to the original problem that we are really interested in. Nevertheless, ϕ^* does offer interesting intuitions that help us construct a scalable, near-optimal heuristic policy applicable to the original problem. In the following, we explain and construct this heuristic policy and prove its equivalence to PAS.

For $j \in \mathcal{J}$, $s \in \mathcal{S}_j$, $n \in \mathcal{C}_{k(s)}$, let

$$\nu_{j,k(s)}^*(n) = 1 - e^{-\frac{\varepsilon_{k(s)} - \varepsilon_{k(s)}^0}{\mu_{k(s)}}}. \quad (20)$$

The optimal policy ϕ^* described in (18) implies the priorities of different servers: for a given ν , if job-server pair (j, s) has $\alpha_{j,s}^{\phi^*}(n) = 1$, $n \in \mathcal{C}_{k(s)}$, then any other pairs (j, s') with $\nu_{j,k(s')}^*(n') > \nu_{j,k(s)}^*(n)$ must have $\alpha_{j,s'}^{\phi^*}(n') = 1$, $n' \in \mathcal{C}_{k(s')}$. The value of $\nu_{j,k(s)}^*(\cdot)$ can be interpreted as the server's potential profits (or subsidy following the idea in [26]) gained by choosing this server s to serve a job of type j if this server is not fully occupied.

A similar property was defined in [26] for a Restless Multi-Armed Bandit Problem (RMABP) and referred to as *Whittle indexability*. When job sizes are exponentially distributed, $J = 1$ and neglecting (5), our problem reduces to a RMABP, and Whittle indices are given by $\nu_{j,k(s)}^*(\cdot)$. There may be no simple closed form for the Whittle indices in the general case with general job-size distributions. Relevant work about RMABP and Whittle indices have been mentioned in Section II.

In the general case, if we always assign incoming jobs of type j to servers with the highest $\nu_{j,k(s)}^*(\cdot)$, for the highest potential profits, among those in the available set \mathcal{S}_j and with vacancies in their buffers, then the resulting policy coincides with PAS described in Section V. PAS is applicable to the original problem, defined by (3)-(5). We discuss PAS in Section VI-B by comparing it to policy ϕ^* described in Proposition 1. If ϕ^* is optimal for the relaxed problem, then the performance of ϕ^* is an upper bound of that of the original problem. If PAS's performance again coincides with ϕ^* , then PAS is optimal for the original problem.

B. Convergence in Performance

1) *Stochastic Processes with Smooth Trajectories*: Let states $n \in \mathcal{C}_k$ for all $k \in \mathcal{K}$ be ordered according to descending values $\nu_{j,k}^*(n)$, where uncontrollable states $n \in \mathcal{U}_k$, for all $k \in \mathcal{K}$, follow the controllable states in the ordering, with $\alpha_{j,s}^{\phi^*}(n) = 0$ for $n \in \mathcal{U}_k$, $s \in \mathcal{R}_k$, $k \in \mathcal{K}_j$, $j \in \mathcal{J}$. Then we place the state $n \in \mathcal{B}_0$ of zero-reward servers, also a controllable state, after all other controllable states but preceding the uncontrollable states. To indicate the order of states, the position of a state in the ordering $i = 1, 2, \dots, I$, where $I = \sum_{k \in \mathcal{K} \cup \{0\}} |\mathcal{B}_k|$, is regarded as its label. Define $\tilde{\mathcal{B}} := \{1, 2, \dots, I\}$. Let n_i represent the server state labeled

by i (i.e., the i th state), and k_i represent the only server group with $n_i \in \mathcal{B}_{k_i}$.

Since each $i \in \tilde{\mathcal{B}}$ is associated with a server group k and a state in \mathcal{B}_k , servers are thus distinguishable only through their current state $i \in \tilde{\mathcal{B}}$.

Let $\mathbf{Z}^\phi(t) = (Z_i^\phi(t) : i \in \tilde{\mathcal{B}})$, and \mathcal{Z} represent the space of all probability vectors of length I . The random variable $Z_i^\phi(t)$ represents the proportion of servers in state $i \in \tilde{\mathcal{B}}$ at time t under policy $\phi \in \Phi$, $t \geq 0$: that is,

$$Z_i^\phi(t) := \frac{1}{|\mathcal{S}| + R_0} |\{s \in \mathcal{S} \mid N_s^\phi(t) = i\}|.$$

Define a mapping m by $m(\mathbf{N}^\phi(t)) = \mathbf{Z}^\phi(t)$. Recall that our server farm is assumed to be empty at $t = 0$, and, correspondingly, define $\mathbf{z}^0 := \mathbf{Z}^\phi(0) = m(\mathbf{0})$. On the arrival or departure of jobs at time t , $\mathbf{Z}^\phi(t)$ transitions to $\mathbf{Z}^\phi(t) + \mathbf{e}_{i,i'}$, where $\mathbf{e}_{i,i'}$ is a vector of which the i th element is $+1/(|\mathcal{S}| + R_0)$, the i' th element is $-1/(|\mathcal{S}| + R_0)$ and otherwise is zero, $i, i' \in \tilde{\mathcal{B}}$. Servers in server group k only appear in state $n_i \in \mathcal{B}_k$; that is, the transition from $\mathbf{Z}^\phi(t)$ to $\mathbf{Z}^\phi(t) + \mathbf{e}_{i,i'}$, $n_i \in \mathcal{B}_k$, $n_{i'} \in \mathcal{B}_{k'}$, $k, k' \in \mathcal{K} \cup \{0\}$, $k \neq k'$ never occurs.

Let $R_k = R_k^0 h$ and $\lambda_j = \lambda_j^0 h$, where $h = 1, 2, \dots$ is called the *scaling parameter*. Correspondingly, let $S_0 := (|\mathcal{S}| + R_0)/h$, then $S_0 \in \mathbb{N}^+$. Since servers in the same state $i \in \mathcal{B}$ are indistinguishable, we define the probability of selecting/activating a server in state i for an arriving job of type j , i.e., the probability of setting $\alpha_{j,s}^{\phi^*}(i) = 1$, as $u_{j,i}^{\phi,h}(\mathbf{z})$, when policy ϕ is used, and $\mathbf{Z}^\phi(t) = \mathbf{z}$. Clearly, $u_{j,i}^{\phi,h}(\mathbf{z}) = 0$ and $\mathbf{z} \in \mathcal{Z}$, if i represents an uncontrollable state.

We obtain, for $i \in \tilde{\mathcal{B}}$, $j \in \mathcal{J}$, $h \in \mathbb{N}^+$, $\mathbf{z} \in \mathcal{Z}$ with $z_i > 0$,

$$u_{j,i}^{\text{PAS},h}(\mathbf{z}) = \min \left\{ 1, \frac{1}{z_i} \max \left\{ 0, \frac{1}{h S_0} - \sum_{i' < i, k_{i'} \in \mathcal{K}_j} z_{i'} \right\} \right\}.$$

We define without loss of generality the transition caused by an arrival event in state i as the transition from state i to state $i + 1$, if $n_i, n_{i+1} \in \mathcal{C}_k$, $k \in \mathcal{K}$. Such a transition from i to $i + 1$ is caused by an arrival of a job of a specified type.

Following the ideas of [21], [28], [38], we then obtain a corollary of [38, Proposition 5] as follows.

Corollary 1. *When the job sizes are exponentially distributed, for any $\delta > 0$, there exists a $\mathbf{z}^{\text{PAS}} \in \mathcal{Z}$ such that*

$$\lim_{h \rightarrow +\infty} \lim_{t \rightarrow +\infty} \frac{1}{t} \int_0^t \mathbb{P} \left\{ \|\mathbf{Z}^{\text{PAS},h}(u) - \mathbf{z}^{\text{PAS}}\| > \delta \right\} du = 0, \quad (21)$$

with given $\mathbf{Z}^{\text{PAS},h}(0) = \mathbf{z}^0$.

Equation (21) indicates that stochastic process $\mathbf{Z}^{\text{PAS},h}(t)$ will go into a close neighborhood of point \mathbf{z}^{PAS} as the scaling parameter h tends to infinity, where the process transition rates of leaving and entering each of its states must be equivalent. Also, since priorities of states are driven by $u_{j,i}^{\text{PAS},h}(\cdot)$, if we start from an empty server farm, the trajectory of $Z_i^{\text{PAS},h}(t)$, $t \geq 0$, is independent from the trajectories of $Z_{i'}^{\text{PAS},h}(t)$, $t \geq 0$, for states $i' > i$ (that is, states with lower priorities). We can then calculate the value of \mathbf{z}^{PAS} from the first element to the last, which coincides with the calculating procedure

of $\lim_{h \rightarrow +\infty} \lim_{t \rightarrow +\infty} \mathbb{E}[\mathbf{Z}^{\phi^*, h}(t)]$ under ϕ^* with the state priorities driven by action variables described in (18) and (19).

Given the reward rate functions for all states $i \in \mathcal{B}$, the long-run average reward $\Gamma^{\phi}(\mathbf{f}^r)$ (the objective function of our problem) of policy ϕ is linear in the expected value $\lim_{t \rightarrow +\infty} \mathbb{E}[\mathbf{Z}^{\phi, h}(t)]$. In other words, when $\mathbf{Z}^{\text{PAS}, h}(t)$ approaches \mathbf{z}^{PAS} in the asymptotic regime, $\Gamma^{\text{PAS}}(\mathbf{f}^r)$ also approaches $\Gamma^{\phi^*}(\mathbf{f}^r)$ for any given $e^* \in \mathbb{R}$. As mentioned at the end of Section VI-A, if ϕ^* is also optimal for the relaxed problem, then PAS is asymptotically optimal in the original problem, because the maximized energy efficiency of the relaxed problem is always an upper bound of that of the original one.

In consequence, if $J = 1$ or Condition 1 is satisfied, Proposition 1 and Corollary 1 yield the asymptotic optimality of PAS as $h \rightarrow +\infty$ in terms of energy efficiency, when job sizes are exponentially distributed.

2) *Bounded Performance Deviation*: According to our result on asymptotic optimality stated in Section VI-B1, the performance deviation between PAS and the optimal solution in the asymptotic regime is directly related to the supremum of Euclidean distances between $\lim_{t \rightarrow +\infty} \mathbb{E}[\mathbf{Z}^{\text{PAS}, h}(t)]$ and \mathbf{z}^{PAS} .

Proposition 2. *When the job sizes are exponentially distributed, for any $\delta > 0$, there exist $\mathbf{z}^{\text{PAS}} \in \mathcal{Z}$, $s > 0$ and $H > 0$, such that, for any $h > H$,*

$$\lim_{t \rightarrow +\infty} \frac{1}{t} \int_0^t \mathbb{P}\left\{\|\mathbf{Z}^{\text{PAS}, h}(u) - \mathbf{z}^{\text{PAS}}\| > \delta\right\} du \leq e^{-sh}, \quad (22)$$

where $\mathbf{Z}^{\text{PAS}, h}(0) = \mathbf{z}^0$.

The proof of Proposition 2 is given in Appendix B. That is, the deviation bound of PAS diminishes exponentially in the scaling parameter h .

As mentioned in Section I, since the asymptotic regime can never be achieved in the real world, Proposition 2 sharpens the asymptotic optimality result: PAS approaches asymptotic optimality very quickly (exponentially) as the problem size increases. Simulation results will be provided in Section VII.

Proposition 2 applies in systems with much more general power functions, extending asymptotic optimality of PAS to more general cases. For instance, by [38, Proposition 1] and Proposition 2, the PAS family is also asymptotically optimal when the service and energy consumption rates of each server is linearly increasing in the number of jobs there, although the linearity is not appropriate in modeling power consumption in Cloud environments. As mentioned in Section II, the problem studied here is already sufficiently complex to prevent existing methods from being applied directly. More general power functions will presumably and significantly complicate the system model and notational definitions and is outside the scope of this paper.

VII. NUMERICAL RESULTS

Here, we numerically demonstrate the effectiveness of PAS by comparing it with the optimal energy efficiency, as a benchmark, in different scenarios with randomly generated server farm systems. We recall that, as mentioned in Sections I and II, the complexity of the server farm model prevents applicability of existing scheduling policies from being direct.

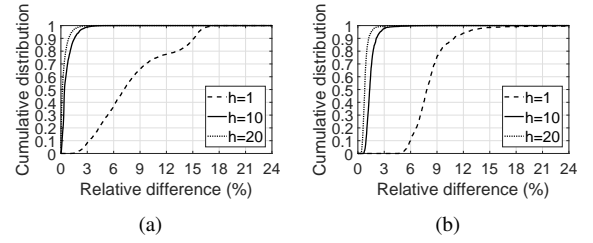


Fig. 1. Cumulative distribution of the normalized performance deviation of the PAS policy: (a) single job type; (b) multiple job types.

In our simulation results, the 95% confidence intervals based on the Student t -distribution are maintained within $\pm 3\%$ of the observed mean. In Sections VII-A and VII-B, we set job-sizes to be exponentially distributed, and in Section VII-C, more realistic job-size distributions are discussed.

A. Deviation Studies

We are interested in the performance deviation of the PAS policy; that is, how large the system should be to guarantee a performance deviation with reasonable bounds. For the sake of simplicity, let OPT represent an optimal solution for the relaxed problem defined by (10), (11), (13) and (14) in the asymptotic regime, we define the *normalized performance deviation* of a policy $\phi \in \Phi$, to be

$$\left(\mathcal{L}^{\text{OPT}}/\mathcal{E}^{\text{OPT}} - \mathcal{L}^{\phi}/\mathcal{E}^{\phi}\right) / \left(\mathcal{L}^{\text{OPT}}/\mathcal{E}^{\text{OPT}}\right).$$

Note that energy efficiency under OPT is an upper bound for that under an optimal solution of the original problem defined by (3)–(5) in the asymptotic regime. Because of the extremely high computational complexity of the original problem, we use OPT as a benchmark in our numerical experiments.

1) *Stochastically Identical Jobs*: We start with the simple case of only one job type ($J = 1$). Consider a server farm with five server groups ($K = 5$), each of which has $R_k^0 = 1$, $k \in \mathcal{K}$, servers when the scaling parameter $h = 1$. Let the buffer sizes B_k of all servers equal 2, for all $k \in \mathcal{K}$. In Figure 1(a), we depict the cumulative distribution of the normalized performance deviation of PAS in terms of energy efficiency, with randomly generated service rates, energy consumption rates and sets of available servers as follows.

- Service rates μ_k , $k \in \mathcal{K}$, are randomly uniformly generated in the range $[1, 10]$;
- Energy efficiencies of servers in the first group are normalized to be 1, i.e., $\mu_1/\varepsilon_1 = 1$; those in successive groups are obtained by randomly uniformly generating the ratio of server energy efficiencies for successive groups, i.e., $(\mu_k/\varepsilon_k)/(\mu_{k-1}/\varepsilon_{k-1})$, $k = 2, 3, \dots, K$, from $[0.5, 1]$ iteratively;
- According to the service rates and energy efficiencies of servers for different groups, we obtain the busy power consumption for all servers, and set the idle power of servers in group k , $k \in \mathcal{K}$, to be $\varepsilon_k \cdot (0.1 + 0.1 \cdot k)$;
- All servers in the server farm are available for incoming jobs $\mathcal{K}_1 = \mathcal{K}$; and
- The average arrival rate λ_1 is set to $\rho \cdot \sum_{k=1}^K \mu_k$ with a given *normalized offered traffic* ρ .

In Figure 1(a), the value of the normalized performance deviation of the PAS policy is decreasing in h , $h = 1, 10, 20$,

and, for all our simulations, is within 3% (of the energy efficiency under OPT) when $h = 20$; that is, twenty servers in each server group. In other words, in this experiment, the PAS policy is already close to OPT when the server farm is relatively small. These results are consistent with the deviation upper bound described by (22). That is, PAS is demonstrated to be near-optimal since the scaling parameter is relatively small, and, in line with our theoretical results (22), is likely to be near optimal for any larger h .

2) *Multiple Job Types*: For the general case of multiple job types, we define servers with the same settings as those for Figure 1(a) except here we set $B_k = 1$ for $k \in \mathcal{K}$. We plot the cumulative distribution of normalized performance deviation of the PAS policy in Figure 1(b), where three different job types have been considered ($J = 3$). The parameters for a job of type $j \in \mathcal{J}$, are generated by:

- we firstly generate a random number m_j of server groups for available servers, following a uniform distribution within $\{1, 2, \dots, K\}$;
- then we randomly pick m_j server groups from the total K ones as the server groups for available servers, and generate the set \mathcal{K}_j of these server groups;
- set the average arrival rate of jobs of type j to be the product of ρ and the sum of service rates of all its available servers, where ρ is given.

We compare PAS with OPT in Figure 1(b) where the heavy traffic condition (Condition 1) is satisfied, so that an optimal solution for the relaxed problem (OPT) exists in the form of (18) and (19). Note that this heavy traffic condition is not necessary for the $J = 1$ case discussed in Section VII-A1. For the general case with $J > 1$, if the heavy traffic condition is not valid, OPT does not necessarily exist in the form of (18) and (19) and it remains unclear how to calculate such an OPT within a reasonable time.

In Figure 1(b), the normalized performance deviation of PAS maintains a trend similar to that in Figure 1(a), decreasing quickly with increasing h , $h = 1, 10, 20$. The normalized performance deviation of PAS is no more than 3% in almost all experiments for Figure 1(b) when $h = 20$. This is consistent with our argument regarding Figure 1(a) that PAS is close to OPT even for a relatively small system and so for such a system with any larger scaling parameter h .

We pick up two specific runs of simulations for Figures 1(a) and 1(b) as examples for both cases, and, in Figure 2, demonstrate the normalized performance deviation of PAS against the scaling parameter h of the server farm for two cases: *Case I* and *Case II* stand for systems with stochastically identical jobs and multiple job types, respectively. The detailed parameter values, which are generated randomly, for Figure 2 are provided in Appendix C.

In Figure 2, the normalized deviation of PAS is seen to approach 0 as h increases, being greater than 1% for $h \geq 26$, for both cases. Figure 2 has a plot of the normalized performance deviation of PAS against the scaling parameter h , with the y -axis in log scale. The curve for Case I appears almost linear in h , and that for Case II convex in h , with an almost linear tail. These results are consistent with the exponentially decreasing upper bound of PAS performance

deviation described by (22). The straight curve for Case I and the straight tail for Case II suggest that the upper bound shown in (22) is likely to be tight.

All the demonstrated simulations with randomly generated parameters have shown convergence between PAS and OPT in energy efficiency since the scaling parameter h is relatively small, implying the near-optimality of PAS for any larger h . PAS is thus appropriate for server farms with realistic scales; that is, large but not necessarily in the asymptotic regime.

B. Case Studies

We now consider the performance of PAS with respect to Google cluster traces of job arrivals in 2011 [39], [40]. The cluster consists of 12.5 thousand machines with arriving jobs classified into four groups. The job arrival rates, estimated as the number of arrived jobs per second, averaged in each hour are plotted in Figure 3.

In Figures 4 and 5, we demonstrate the effectiveness of the PAS policy by comparing it with a baseline policy, Join-the-Shortest-Queue (JSQ). JSQ is a load balancing policy that is proved to maximize the number of processed jobs within a given time period [41].

Unlike in the simulations presented in Section VII-A, in this subsection, we do not assume Poisson arrival process, so that the different service disciplines potentially lead to different steady state distributions or the long-run average performance of the system. Consider two classical service disciplines for the simulation results in this subsection and Section VII-C: the PS and the Shortest-Remaining-Processing-Time (SRPT) disciplines. The PS discipline of processing jobs is appropriate for web server farms to avoid unfair processing delays between jobs, especially when their job sizes are highly varied [42], [43]. The SRPT is a well-known discipline that minimizes the mean response time [44].

There are ten server groups each of which contains 1.25 thousand servers with randomly generated service and power consumption rates and availability for serving jobs of different types. The detailed parameter values for simulations in this subsection are provided in Appendix D.

Observing Figures 4 and 5, for either service discipline, PAS achieves clearly higher energy efficiency while maintaining comparable job throughput with those of JSQ. For the PAS policy, the energy efficiency and job throughput curves for SRPT are slightly higher than those for PS in terms of both energy efficiency and job throughput. This is because SRPT is a discipline aiming at load balancing while PS is designed for guaranteeing fairness and robustness.

Moreover, for the simulations in Figures 4 and 5, although the traffic intensity during the peak hours is higher than one (heavy traffic condition is satisfied), the simulated number of blocked jobs is zero for both PAS and JSQ. Because the scale of the entire server farm is sufficiently large, with 12.5 thousand servers, and the peak hours with heavy traffic are relatively few as shown in Figure 3, there are sufficiently many buffer slots in the server farm to digest the heavy traffic during peak hours and thus the number of blocked jobs is negligible in the presented simulations.

Also, we tested the blocking probability of the Google trace-logs used in Figure 5. The system parameters are the same as

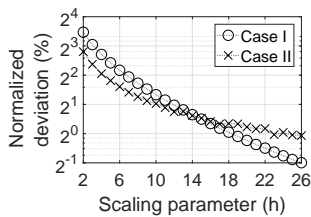


Fig. 2. Performance of the PAS policy against the scaling parameter.

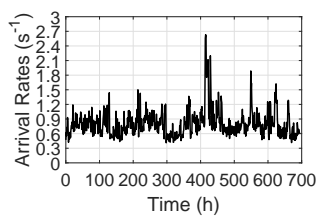


Fig. 3. Arrival rates of Google trace logs.

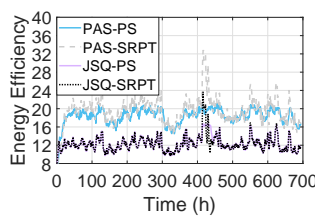


Fig. 4. Energy efficiency of PAS and JSQ.

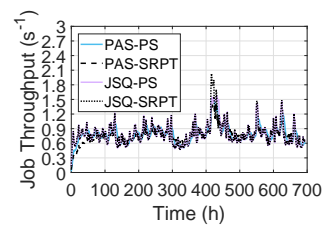


Fig. 5. Job throughput of PAS and JSQ.

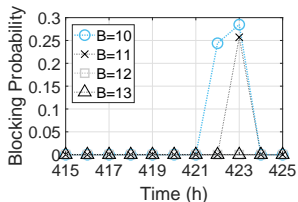


Fig. 6. Blocking probability of PAS under PS during the peak hours.

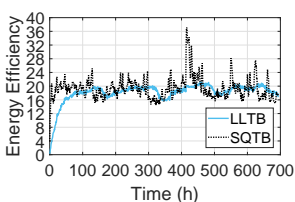


Fig. 7. Energy efficiency of PAS with different tie-breaking rules.

those for the simulations presented in Figure 5, except that the server buffer sizes are set to B and the B takes different values: 10, 11, 12, 13. For all the tested buffer sizes, PAS under SRPT and JSQ under both disciplines incur zero blocked jobs based on our simulations; while, as demonstrated in Figure 6, PAS under PS incurs non-negligible blocking probabilities during the peak hours for $B = 10, 11$ and this reduces to zero for $B \geq 12$. These results strengthen our earlier argument: when the total buffer size of the entire server farm is sufficiently large, the number of blocked jobs becomes negligible even if the heavy traffic condition is satisfied. For a large server farm with 12.5 thousand servers, such as the Google cluster mentioned above, the total buffer size of the entire server farm is already large with relative small B .

Moreover, to complete the discussion, in Figure 7, we plot the energy efficiency of PAS under PS with a different tie-breaking rule, but the same settings as the simulations presented in Figure 5. Recall that all our theoretical results apply to any tie-breaking rule, and, as described in Section V, we have just chosen the simplest: when there is more than one server with the highest effective energy efficiency, assign jobs to the server with lowest label. We refer to it as *Lowest Label Tie Breaking (LLTB)*. Alternatively, for multiple servers with the same effective energy efficiency, we could choose the one with the shortest queue; this is referred to as *Shortest Queue Tie Breaking (SQTb)*.

We can see from Figure 7 that LLTB achieves flatter performance, while SQTb is more variable. But in reality there appears to be no general advantage of one over the other. In particular, SQTb outperforms LLTB by around 0.5% with respect to the total energy efficiency and both cases incur no blocked jobs. We then argue that PAS under PS is not very sensitive to different tie-breaking rules. The robustness of PAS to service disciplines demands further exploration involving more comprehensive case studies with real-world trace-logs, but that is outside the scope of this paper.

C. Robustness Studies

In practice, job duration times for many online applications have been studied and known to be characterized by heavy

tailed distributions [45], which is at odds with our exponential assumption. Hence, it is important to understand the sensitivity of the PAS policy to different job-size distributions. In this context, we consider two heavy-tailed distributions: Pareto with shape parameter 2.001 (Pareto-F for short) and Pareto with shape parameter 1.98 (Pareto-INF for short); these are set to have unit mean. Note that Pareto-F and Pareto-INF are Pareto distributions with finite and infinite variance, respectively.

With the same settings as for Section VII-A2, here, we test the energy efficiency of PAS with exponentially, Pareto-F and Pareto-INF distributed job sizes. We also consider a case where the job sizes of different types are distributed differently; this is referred to as the *mixed* case.

In Figure 8, we demonstrate the robustness of PAS under the PS and SRPT disciplines with respect to different job size distributions. Let Γ^D represent the energy efficiency of the server farm under PAS with job-size distribution D , where $D = \text{exponential, mixed, Pareto-F or Pareto-INF}$.

In Figure 8, we show the cumulative distribution of $(\Gamma^D - \Gamma^{\text{exponential}})/\Gamma^{\text{exponential}}$; that is, the relative difference of energy efficiency with job size distribution D from the one with exponentially distributed job sizes. In Figure 8(a), this relative difference is within $\pm 1\%$ in all our experiments with randomly generated parameters; while, varies between -1% and 3% in Figure 8(b). PAS is resilient to these tested job-sizes distributions under PS and SRPT, although SRPT incurs slightly higher variance than PS.

VIII. CONCLUSIONS

We have studied the job-assignment problem in a server farm model consisting of a large number of abstracted servers that are possibly diverse in service rates, energy consumption rates, buffer sizes (service capacities) and the ability to serve different jobs. Also, as described in Section III, in this work, the relationship between energy consumption and service rates of servers (abstracted computer components) can be arbitrary, and be determined by the functional features and profiles of servers. By assigning jobs to efficient servers, we aim to balance the job throughput and the power consumption of the system; that is, we aim to maximize the energy efficiency defined as the ratio of long-run average job departure rate to the long-run average energy consumption rate of the entire server farm system.

Following the idea of Whittle relaxation [26], we have proposed the scalable policy, PAS, that prioritizes servers according to only intrinsic attributes and binary state information of different servers among the available ones. PAS accounts

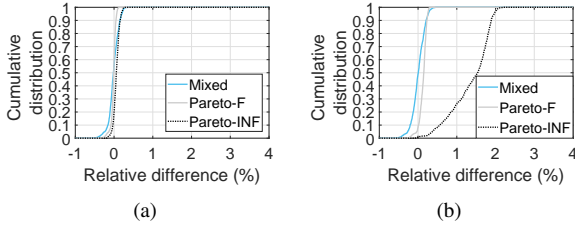


Fig. 8. Cumulative distribution of the relative difference of Γ^D to $\Gamma^{\text{exponential}}$ with $D = \text{mixed, Pareto-F or Pareto-INF}$ under (a) the PS discipline; and (b) the SRPT discipline.

for the availability of servers to service different jobs, enabling the applicability of our server farm model to geographically separated computing systems.

To the best of our knowledge, there is no existing work that proposes scalable, infinite horizon policies for such heterogeneous server farms at realistic scales, with a rigorous analysis of performance deviation in terms of energy efficiency.

We have proved that, when job sizes are exponentially distributed, if the blocking probabilities of jobs are always positive or $J = 1$, there exists a deviation bound for PAS which is exponentially decreasing in the number of servers in server groups and the average arrival rates of jobs proportionately. This deviation bound indicates the asymptotic optimality of PAS, and, more importantly, significantly improves the asymptotic optimality results: PAS approaches asymptotic optimality very quickly (exponentially) as the server farm size increases. Numerical results illustrate that PAS is already close to OPT for only 100 servers, consistent with our deviation bound. We infer that PAS is nearly optimal for even a relatively small system and any larger one. The robustness of PAS to three different job-size distributions has been tested numerically, with resulting values of energy efficiency in our simulations close to those with exponentially distributed job sizes.

APPENDIX A PROOF OF PROPOSITION 1

Fix a server, $s \in \mathcal{S}$, starting in state $n_s \in \mathcal{B}_{k(s)}$, a policy $\phi \in \Phi_s$, and a reward rate $f_{k(s)} \in \mathbb{R}^{\mathcal{B}_{k(s)}}$. Write $V_s^\phi(n_s, f_{k(s)})$, for the expected value of the cumulative reward for server s that ends when it first enters an absorbing state $n_s^0 \in \mathcal{B}_{k(s)}$. In particular, $V_s^\phi(n_s^0, f_{k(s)}) = 0$ for any $\phi \in \Phi_s$. We assume, without loss of generality, that $n_s^0 = 0$ for all $s \in \mathcal{S}$. Note that such a $V_s^\phi(n_s, f_{k(s)})$ is dependent on the values of ν and γ , though we refrain from including them as superscript/subscript to simplify notation. Let $V_s(n_s, f_{k(s)}) = \max_{\phi \in \Phi_s} V_s^\phi(n_s, f_{k(s)})$, $f_{k(s)} \in \mathcal{R}_{k(s)}$, $s \in \mathcal{S}$, $n_s \in \mathcal{B}_{k(s)}$.

Now, let \mathcal{P}_s^H , $s \in \mathcal{S}$, represent a process for server s that starts from state 0 until it reaches state 0 again, where $\phi \in \Phi_s$ is constrained to those policies satisfying $\sum_{j \in \mathcal{J}_s} \alpha_{j,s}^\phi(0) > 0$. From [46, Corollary 6.20 and Theorem 7.5], when the job sizes are exponentially distributed, the process \mathcal{P}_s^H is a renewal interval of the long-run process, so that the average reward of \mathcal{P}_s^H equals the long-run average reward of the process for server s under the same policy.

Define, for $s \in \mathcal{S}$, $\phi \in \Phi_s$, $n \in \mathcal{B}_{k(s)}$,

$$\tilde{f}_{k(s)}(n) = \begin{cases} f_{k(s)}^r(n) - \sum_{j \in \mathcal{J}_s} \nu_j \alpha_{j,s}^\phi(n), & \text{if } n \in \mathcal{C}_{k(s)}, \\ f_{k(s)}^r(n) - \sum_{j \in \mathcal{J}_s} (\nu_j + \eta_{j,s}) \alpha_{j,s}^\phi(n), & \text{otherwise,} \end{cases}$$

where $f_{k(s)}^r(n)$ is as defined in (9). Following [46, Theorem 7.6, Theorem 7.7] and [21, Corollary 1], there exists $g \in \mathbb{R}$, with $\tilde{f}_{k(s)}^g(n) = f_{k(s)}^r(n) - g$, $n \in \mathcal{B}_{k(s)}$, such that if policy $\phi^* \in \Phi_s^H$ maximizes the expected cumulative reward of process \mathcal{P}_s^H with reward rate $\tilde{f}_{k(s)}^g(n)$, then ϕ^* also maximizes the long-run average reward of server s with reward rate $f_{k(s)}^r(n)$ among all policies in Φ_s^H . This value of g , denoted by g_s^* , is just the maximized long-run average reward.

For $s \in \mathcal{S}$, $j \in \mathcal{J}_s$, $n \in \mathcal{C}_{k(s)}$ again for notational simplicity, we write $V_s^g(n) = V_s(n, f_{k(s)}^g)$ and $r_s^g(n) = f_{k(s)}^r(n) - g$. Note that, in this context, $V_s^g(0) = V_s(0, \tilde{f}_{k(s)}^g) = 0$ since state 0 is the absorbing state.

Lemma 1. *When the job sizes are exponentially distributed, there exists a policy $\phi^* \in \Phi_s$, $s \in \mathcal{S}$, that maximizes the objective function given by (16), satisfying: for $n \in \mathcal{C}_{k(s)}$,*

$$\alpha_{j,s}^{\phi^*}(n) = \begin{cases} 1, & \text{if } \frac{\nu_j}{\lambda_j} < V_s^{g^*}(n+1) - V_s^{g^*}(n), \\ 1 \text{ or } 0, & \text{if } \frac{\nu_j}{\lambda_j} = V_s^{g^*}(n+1) - V_s^{g^*}(n), \\ 0, & \text{otherwise,} \end{cases} \quad (23)$$

where $g^* := g_s^*$ for notational simplicity.

Proof: We start from (23). Let $\bar{\lambda}_{j,s}(n) = \sum_{j' \in \mathcal{J}_s: j \neq j'} a_{j',s}^{\phi^*}(n=1) \lambda_{j'}$. Then, from the Bellman equation for Markov Decision Process (MDP), we obtain (23) for $n \in \mathcal{C}_{k(s)} \setminus \{0\}$, which takes values independent of $\bar{\lambda}_{j,s}(n)$ and identically for all $j \in \mathcal{J}_s$.

Similarly, the maximization problem based on the Bellman equation for state $n = 0$ lead to, for $\bar{\lambda}_{j,s}(0) > 0$, equation (23) when g is set to be g^* . For $\bar{\lambda}_{j,s}(0) = 0$, there exists an optimal policy $\phi^* \in \Phi_s$ with $a_{j,s}^{\phi^*}(0) = 1$ if and only if $V_s^{g^*}(1) - (e^* \varepsilon_{k(s)}^0 + \nu_j + g^*) / \lambda_j = 0$, and $g^* \geq -e^* \varepsilon_{k(s)}^0$, since g^* equals the optimal average reward of the same \mathcal{P}_s^H with reward rate $\tilde{f}_{k(s)}(n)$, $n \in \mathcal{B}_{k(s)}$. These two equations lead to (23) with $n = 0$. The lemma is proved. ■

Lemma 2. *When the job sizes are exponentially distributed, if $\nu = \nu_j / \lambda_j$ for all $j \in \mathcal{J}$, there exists a policy $\phi^* \in \Phi_s$ ($s \in \mathcal{S}$) maximizing the objective function given by (16), and satisfying (18) for $s \in \mathcal{S}_j$ and $n \in \mathcal{C}_{k(s)}$.*

Proof: From Lemma 1, for any $s \in \mathcal{S}_j$, the maximization problem defined in (16) is equivalent to

$$\max_{m \in \mathcal{C}_{k(s)}} \lambda \left(\frac{\mu_{k(s)} - e^*(\varepsilon_{k(s)} - \varepsilon_{k(s)}^0) - \nu}{\mu_{k(s)}} \right)^{\frac{\sum_{n=0}^m \left(\frac{\lambda}{\mu_{k(s)}} \right)^n}{\sum_{n=0}^{m+1} \left(\frac{\lambda}{\mu_{k(s)}} \right)^n}},$$

where $\lambda = \sum_{j \in \mathcal{J}_s} \lambda_j$. Solving it, we obtain (18). ■

We now give the proof of Proposition 1.

Proof: The proof consists of two parts: label=(α^*)

- 1) we construct a set of ν , γ , $\varphi_j \in \bar{\Phi}_j$, $j \in \mathcal{J}$, and $\phi_s^* \in \Phi_s$, $s \in \mathcal{S}$, where ϕ_s^* and φ_j maximize the objective functions defined in (16) and (17), respectively;
- 2) and we prove that a policy consisting of such φ_j , $j \in \mathcal{J}$, and ϕ_s^* , $s \in \mathcal{S}$, maximizes the relaxed problem defined by (10), (11), (13)-(14).

We firstly discuss the case where Condition 1 holds. Let

- 1) $\nu_j/\lambda_j = \nu$ and $\nu = \min\left\{0, \min_{s \in \mathcal{S}_j} \left(1 - e^{*\frac{\varepsilon_{k(s)} - \varepsilon_{k(s)}^0}{\mu_{k(s)}}}\right)\right\}$;
- 2) $\gamma_j = -\lambda_j \nu$, $j \in \mathcal{J}$.

From Lemma 2, there is an optimal policy $\phi_s^* \in \Phi_s$ that maximizes the problem defined in (16), satisfying $\alpha_{j,s}^{\phi_s^*}(n) = 1$ for all $j \in \mathcal{J}$, $s \in \mathcal{S}_j$, $n \in \mathcal{C}_{k(s)}$. There is also a policy $\varphi_j \in \overline{\Phi}_j$, $j \in \mathcal{J}$, for the maximization problem defined by (17), satisfying $\overline{\alpha}_j^{\varphi_j} = 1 - A_j$ and $\alpha_j^{\varphi_j} = 1 - A_j$. Note that $1 - A_j \in [0, 1)$ under Condition 1.

Therefore, Constraints (11), (13) and (14) are satisfied with equality: the complementary slackness condition for the dual and primal problems is satisfied. The optimal solution ϕ^* determined by ϕ_s^* ($s \in \mathcal{S}$) and φ_j ($j \in \mathcal{J}$) that maximizes the Lagrangian problem defined in (15) will also maximize the primal problem defined by (10), (11), (13)-(14).

We now consider the case where $J = 1$ and denote the only j in \mathcal{J} by j^* . The proposition can be proved along the same lines as the $J > 1$ case if $A_{j^*} \leq 1$. If $A_{j^*} > 1$, then let $\gamma_{j^*} = \max\{0, -\nu_{j^*}\}$. From Lemma 2, there is an optimal policy $\phi_s^* \in \Phi_s$ that maximizes the problem defined in (16), satisfying (18) for $j = j^*$, $s \in \mathcal{S}_{j^*}$ and $n \in \mathcal{C}_{k(s)}$. Since $A_{j^*} > 1$, there exists a $\nu_{j^*} \geq \min_{s \in \mathcal{S}_{j^*}} \lambda_{j^*} (1 - e^{*(\varepsilon_{k(s)} - \varepsilon_{k(s)}^0)/\mu_{k(s)}})$, such that $\sum_{s \in \mathcal{S}_{j^*}} \sum_{n \in \mathcal{B}_{k(s)}} \pi_s^{\phi_s^*}(n) \alpha_s^{\phi_s^*}(n) = 1$. Note that ϕ_s^* is dependent on ν_{j^*} .

On the other hand, the setting of γ_{j^*} guarantees that there is a policy $\varphi \in \overline{\Phi}_{j^*}$ maximizing the objective function defined by (17), and satisfying $\alpha_{j^*}^{\varphi} = 0$ and $\Theta(\overline{\alpha}_{j^*}^{\varphi}) = 0 = \Theta(1 - A_{j^*})$, e.g., equations (11) and (13) achieve equality. That is, such a ν_{j^*} , γ_{j^*} , $\phi_s^* \in \Phi_s$, $s \in \mathcal{S}_{j^*}$ and $\varphi \in \overline{\Phi}_{j^*}$, make the complementary slackness condition satisfied. ■

APPENDIX B PROOF OF PROPOSITION 2

Consider $J + |\mathcal{S}|$ sequences of positive reals $t_{s,m}^h$, $m = 1, 2, \dots$, for $s = 1, 2, \dots, J + |\mathcal{S}|$, where $h \in \mathbb{N}_+$ is the scaling parameter. We define $t_{s,m}^h$ for $s = 1, 2, \dots, J$ as time of the m th arrival of jobs of type s , and define $t_{s,m}^h$ for $s = J+1, J+2, \dots, J + |\mathcal{S}|$ to be the time of the m th potential departure of jobs on server labeled by $s - J$. Define $t_{s,0}^h = 0$ for any $s = 1, 2, \dots, J + |\mathcal{S}|$. For our network system, the inter-arrival and inter-departure times are positive with probability 1 and, also with probability 1, no two events occur at the same time. Let $\tau(t)$ represent the latest event, either arrival or potential departure, that happens before time t . We define a random vector ξ_t^h , for $s = 1, 2, \dots, |\mathcal{S}| + J$ and $t \geq 0$: if $\tau(t_{s,m}^h) \leq t < t_{s,m^*}^h$ where the m^* satisfies $t_{s,m^*-1}^h \leq t < t_{s,m^*}^h$, then $\xi_{s,t}^h = 1/(t_{s,m^*}^h - \tau(t_{s,m^*}^h))$; otherwise, $\xi_{s,t}^h = 0$. The sample paths of ξ_t^h are almost surely continuous in $t \geq 0$, except for a finite number of discontinuities of the first kind in a bounded period of $t > 0$. Let $\Lambda_{j,i}(\mathbf{x}) = u_{j,i}^{PAS,h}(\frac{\mathbf{x}}{hS_0}) z_{i,h} S_0$, which is independent from h from the definition of $u_{j,i}^{PAS,h}(\cdot)$. We define a function, $Q^h(i, i', \mathbf{x}, \xi^h)$, for $h \in \mathbb{N}^+$, $n_i, n_{i'} \in \mathcal{B}_k$, $k \in \mathcal{K} \cup \{0\}$, $\mathbf{x} \in \mathbb{R}^I$, $\xi^h \in \mathbb{R}^{hS_0+J}$, by: if $k_i = k_{i'}$ and $i' = i + 1$,

$$Q^h(i, i', \mathbf{x}, \xi^h) = \sum_{j \in \mathcal{J}_{k_i}} \left[\Lambda_{j,i}(\mathbf{x}) + f_{i,a,j}^{0,h}(\mathbf{x}) \right] \xi_j^h;$$

if $k_i = k_{i'}$ and $i' = i - 1$,

$$Q^h(i, i', \mathbf{x}, \xi^h) = \sum_{m=\lceil x_{i-1}^- \rceil+1}^{\lceil x_i^- \rceil} \xi_{m+J}^h + \sum_{m=1}^{hS_0} f_{i,a,m}^h(\mathbf{x}) \xi_{m+J}^h;$$

otherwise, $Q^h(i, i', \mathbf{x}, \xi^h) = 0$; where $\mathcal{J}_k = \{j \in \mathcal{J} \mid k \in \mathcal{K}_j\}$, $k \in \mathcal{K}$, $x_i^- = \sum_{m=1}^i x_m$ with $x_0^- = 0$, and $f_{i,a}^{0,h}(\mathbf{x}) \in \mathbb{R}^J$ and $f_{i,a}^h(\mathbf{x}) \in \mathbb{R}^{hS_0}$ with $0 < a < 1$ are appropriate functions as described in [38] such that Q^h is Lipschitz continuous in \mathbf{x} . For the special case $h = 1$ and for any given $0 < a < 1$, $Q^{\phi,1}(i, i', \mathbf{x}, \xi^1)$ satisfies a Lipschitz condition over $\mathbf{x} \in \mathbb{R}^I$ and $\xi^1 \in \mathbb{R}^{S_0+J}$. For $0 < a < 1$ and $\varepsilon > 0$, define

$$\begin{aligned} \dot{X}_{t,\varepsilon}^\varepsilon &:= \sum_{i'=1}^I Q^1(i', i, \mathbf{X}_t^\varepsilon, \xi_{t/\varepsilon}^1) - Q^1(i, i', \mathbf{X}_t^\varepsilon, \xi_{t/\varepsilon}^1) \\ &= (\overline{q}_i(\mathbf{X}_t^\varepsilon), \xi_{t/\varepsilon}^1), \end{aligned}$$

where $\overline{q}_i(\mathbf{x})$ is a vector of length $S_0 + J$ dependent on \mathbf{x} and (\cdot, \cdot) represents inner product of vectors. It follows that $\dot{X}_{t,\varepsilon}^\varepsilon$ satisfies a Lipschitz condition over \mathbf{X}_t^ε and $\xi_{t/\varepsilon}^1$. Let

$$b(\mathbf{x}, \xi) := ((\overline{q}_i(\mathbf{x}), \xi) : i = 1, 2, \dots, I) := \overline{Q}(\mathbf{x})\xi,$$

where where $\overline{Q}(\mathbf{x})$ is a $I \times (S_0 + J)$ matrix. For any $\mathbf{x} \in \mathbb{R}^I$, $\delta > 0$, there exists $\overline{b}(\mathbf{x})$ satisfying

$$\lim_{T \rightarrow +\infty} \mathbb{P} \left\{ \left\| \frac{1}{T} \int_t^{t+T} b^\phi(\mathbf{x}, \xi_s^1) ds - \overline{b}(\mathbf{x}) \right\| > \delta \right\} = 0, \quad (24)$$

uniformly in $t > 0$. Let $\overline{\mathbf{x}}(t)$ be the solution of $\dot{\overline{\mathbf{x}}}(t) = \overline{b}(\overline{\mathbf{x}}(t))$ and $\overline{\mathbf{x}}(0) = \mathbf{X}_0^\varepsilon = \mathbf{x}_0$. Define, for $Y = S_0 + J$, the random variables ξ_t^1 and vectors $\alpha_t \in \mathbb{R}^Y$,

$$\int_0^T H_\xi(\alpha_t) dt = \lim_{\varepsilon \rightarrow 0} \varepsilon \ln \mathbb{E} \exp \left\{ \frac{1}{\varepsilon} \int_0^T (\alpha_t, \xi_{t/\varepsilon}^1) dt \right\}. \quad (25)$$

We discuss the existence of $H_\xi(\cdot)$ satisfying (25) next. Let $\Upsilon_T = \int_0^T \xi_t^1 dt$, which is a Poisson distributed random vector. Writing $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_Y)$, we obtain

$$H_\xi(\alpha) = \sum_{j=1}^I \lambda_j (e^{\alpha_j} - 1) + \sum_{s \in \mathcal{S}} \mu_{k(s)} (e^{\alpha_{J+s}} - 1). \quad (26)$$

It follows that $H_\xi(\alpha)$ satisfying (26) is bounded for any bounded α , so that the functional $H_\xi(\alpha_t)$ satisfying (25) also exists for any continuous α_t on $0 \leq t \leq T$. For a vector $\mathbf{U} \in (\mathbb{R}^+)^Y$, define a function $\mathcal{T}^{\mathbf{U}}$ for $i = 1, 2, \dots, Y$ and $\mathbf{x} \in \mathbb{R}^Y$: if $0 \leq x_i \leq U_i$, $\mathcal{T}_i^{\mathbf{U}}(\mathbf{x}) = x_i$; if $x_i > U_i$, $\mathcal{T}_i^{\mathbf{U}}(\mathbf{x}) = U_i$; otherwise, $\mathcal{T}_i^{\mathbf{U}}(\mathbf{x}) = 0$. For $\tilde{\alpha}, \mathbf{x} \in \mathbb{R}^I$, define

$$H(\mathbf{x}, \tilde{\alpha}) = \lim_{T \rightarrow +\infty} \frac{1}{T} \ln \mathbb{E} \exp \left\{ \int_0^T (\tilde{\alpha}, \tilde{b}^{\mathbf{U}}(\mathbf{x}, \xi_t^1)) dt \right\} \quad (27)$$

where $\tilde{b}^{\mathbf{U}}(\mathbf{x}, \mathbf{y}) := b(\mathbf{x}, \mathcal{T}^{\mathbf{U}}(\mathbf{y})) = \overline{Q}(\mathbf{x})\mathcal{T}^{\mathbf{U}}(\mathbf{y})$. Here, since $b(\cdot, \cdot)$ is Lipschitz continuous in both arguments, all of the elements of \mathbf{U} are finite and positive, and for $\overline{Q}(\cdot) = (q_{i,j})$, $|q_{i,j}| < +\infty$ for all $i = 1, 2, \dots, I$ and $j = 1, 2, \dots, Y$, we obtain that $\tilde{b}^{\mathbf{U}}(\cdot, \cdot)$ is bounded and Lipschitz continuous in both arguments. From [47, Lemma 4.1, Chapter 7], $H(\mathbf{x}, \mathbf{y})$ is jointly continuous in both arguments and convex in the second argument.

We obtain from (27), for any $\tilde{\alpha}, \mathbf{x} \in \mathbb{R}^I$, $|H(\mathbf{x}, \tilde{\alpha})| \leq H_\xi(\mathcal{A}(\tilde{\alpha}\overline{Q}(\mathbf{x})))$, where $\mathcal{A}(\mathbf{x})$ takes absolute values of all \mathbf{x} 's elements. Recall that $\overline{Q}(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^I$, is Lipschitz continuous on \mathbf{x} . For any compact set $\mathcal{A}^c \subset \mathbb{R}^I$ and $\tilde{\alpha} \in \mathcal{A}^c$, $H(\mathbf{x}, \tilde{\alpha})$

obtained from (27) is bounded and because of its joint continuity, is Riemann integrable. Hence, the $H(x, \tilde{\alpha})$ defined in (27) satisfies

$$\int_0^T H(x_t, \tilde{\alpha}_t) dt = \lim_{\epsilon \rightarrow 0} \epsilon \ln \mathbb{E} \exp \left\{ \frac{1}{\epsilon} \int_0^T (\tilde{\alpha}_t, \xi_{t/\epsilon}^1) dt \right\}. \quad (28)$$

Now we consider the Legendre transform of $H(x, \tilde{\alpha})$:

$$L(x, \beta) = \sup_{\tilde{\alpha} \in \mathbb{R}^I} [(\tilde{\alpha}, \beta) - H(x, \tilde{\alpha})]. \quad (29)$$

$L(x, \beta)$ is strictly convex in the second argument if $H(x, \tilde{\alpha})$ is strictly convex in the second argument. Let $\tilde{\alpha} = \mathbf{0}$, $(\tilde{\alpha}, \beta) - H(x, \tilde{\alpha}) = 0$, so that $L(x, \beta)$ is always non-negative.

Lemma 3. *If $L(\varphi, \beta)$ is strictly convex in the second argument, then $L(\varphi, \beta) = 0$ if and only if $\beta = \mathbb{E} \tilde{b}^U(\varphi, \xi_t^1)$.*

Proof: The $L(\varphi, \beta) = 0$ when $\beta = \mathbb{E} \tilde{b}^U(\varphi, \xi_t^1)$, [47, Chapter 7, Section 4]. Together with non-negativity and strict convexity of $L(\varphi, \beta)$, for a given $\varphi \in \mathbb{R}^I$, $L(\varphi, \beta) = 0$ if and only if $\beta = \mathbb{E} \tilde{b}^U(\varphi, \xi_t^1)$. ■

The second derivative $\partial^2 H / \partial \tilde{\alpha}^2$ exists and is continuous in U , and as $U \rightarrow \infty$, it converges point-wisely to $\partial^2 H_\xi(\tilde{\alpha} \bar{Q}(x)) / \partial \tilde{\alpha}^2$; the function $H_\xi(\tilde{\alpha} \bar{Q}(x))$ is strictly convex in $\tilde{\alpha}$ by (26) and the second derivative in $\tilde{\alpha}$ exists. Thus, for sufficiently large U , $L(\varphi, \beta)$ is always strictly convex in the second argument.

Proof: Let $S_{0,T}(\varphi) = \int_0^T L(\varphi_t, \dot{\varphi}_t) dt$, where φ denotes a trajectory $\varphi_t \in \mathbb{R}^I$ ($0 \leq t \leq T$), and let $C_{0,T}$ represent the compact set of all such trajectories with $\varphi_0 = x_0 \in \mathbb{R}^I$. Define a closed set $A(U, \delta) := \{\varphi \in C_{0,T} \mid \|\varphi_t - \tilde{x}_t^U\| \geq \delta\}$ where \tilde{x}_t^U is the solution of $\dot{\tilde{x}}_t^U = \mathbb{E} \tilde{b}^U(\varphi, \xi_t^1)$ with $\tilde{x}_0^U = x_0$. From [47, Theorem 4.1 in Chapter 7 & Theorem 3.3 in Chapter 3], for any $U \in (\mathbb{R}^+)^Y$ and $\delta > 0$,

$$\overline{\lim}_{\epsilon \rightarrow 0} \epsilon \ln \mathbb{P} \left\{ \sup_{0 \leq t \leq T} \|\tilde{X}_t^{U,\epsilon} - \tilde{x}_t^U\| > \delta \right\} \leq - \inf_{\varphi \in A(U, \delta)} S_{0,T}(\varphi),$$

where $\tilde{X}_t^{U,\epsilon}$ is the solution of

$$\dot{\tilde{X}}_t^{U,\epsilon} = \tilde{b}^U(\tilde{X}_t^{U,\epsilon}, \xi_t^1), \quad \tilde{X}_0^{U,\epsilon} = x_0.$$

Also, because

$$\lim_{U \rightarrow \infty} \tilde{x}_t^U = \bar{x}(t) \quad \text{and} \quad \lim_{U \rightarrow \infty} \tilde{X}_t^{U,\epsilon} = X_t^\epsilon,$$

by Lemma 3, we obtain, for any $\delta > 0$, there exist $s > 0$ and $\epsilon_0 > 0$ such that, for all positive $\epsilon < \epsilon_0$,

$$\mathbb{P} \left\{ \sup_{0 \leq t \leq T} \|X_t^\epsilon - \bar{x}(t)\| > \delta \right\} \leq e^{-\frac{\delta}{\epsilon}}. \quad (30)$$

Recall that functions Q^h and b are dependent on a parameter $a \in (0, 1)$. Equation (30) holds for any given a . Because of the Lipschitz behavior of \tilde{X}_t^ϵ and $\bar{x}(t)$ on $0 < a < 1$, $\lim_{a \rightarrow 0} d\tilde{X}_t^\epsilon / da = 0$ and $\lim_{a \rightarrow 0} d\bar{x}(t) / da = 0$, Equation (30) also holds in the limiting case $a \rightarrow 0$. By slightly abusing notation, in the following, we still use Q^h , b and $\bar{x}(t)$ to represent $\lim_{a \rightarrow 0} Q^h$, $\lim_{a \rightarrow 0} b$ and $\lim_{a \rightarrow 0} \bar{x}(t)$.

Along similar lines to [21], we interpret the scalar ϵ and the scaling effects in another way. For $x \in \mathbb{R}^I$ and $\xi^h \in \mathbb{R}^{hS_0+J}$, define

$$b^h(x, \xi^h) := \sum_{i'=1}^I Q^h(i', i, x, \xi^h) - Q^h(i, i', x, \xi^h).$$

If we set $\epsilon = 1/h$, then following the same technique as [21], for any $x \in \mathbb{R}^I$, $h \in \mathbb{N}^+$ and $T > 0$, we observe

that $\int_0^T b(x, \xi_{t/\epsilon}^1) dt$ and $\int_0^T (b^h(hx, \xi_t^h)/h) dt$ are identically distributed. Define $Z_0^\epsilon = Z_0^h = x_0/S_0 = z^0$, and

$$\dot{Z}_t^h = \frac{b^h(hS_0 Z_t^h, \xi_t^h)}{hS_0} \quad \text{and} \quad \dot{Z}_t^\epsilon = \frac{b(S_0 Z_t^\epsilon, \xi_{t/\epsilon}^1)}{S_0}.$$

From (30), for any $T > 0$, $\delta > 0$, there exist positive s and H such that, for all $h > H$,

$$\mathbb{P} \left\{ \sup_{0 \leq t \leq T} \|Z_t^h - \bar{x}(t)/S_0\| > \delta \right\} \leq e^{-sh}. \quad (31)$$

Effectively then, scaling time by $\epsilon = 1/h$ is equivalent to scaling system size by h . From (31), Corollary 1 and [38, Lemma 3], for any $T > 0$, $\delta > 0$, there exist $s > 0$ and $H > 0$ such that for any $h > H$, (22) holds. ■

APPENDIX C

SETTINGS FOR SIMULATIONS IN FIGURE 2

Define Case I as a system with same settings as in the simulations for Figure 1(a), except for the following parameters:

- $\mu_1 = 8.06114$, $\epsilon_1^0 = 0.80611$, $\epsilon_1 = 8.06114$;
- $\mu_2 = 4.05127$, $\epsilon_2^0 = 0.94774$, $\epsilon_2 = 4.73868$;
- $\mu_3 = 3.70788$, $\epsilon_3^0 = 2.21086$, $\epsilon_3 = 7.36952$;
- $\mu_4 = 2.88018$, $\epsilon_4^0 = 4.40851$, $\epsilon_4 = 11.02129$;
- $\mu_5 = 2.44950$, $\epsilon_5^0 = 8.79314$, $\epsilon_5 = 17.58627$;
- and $\lambda_1 = 6.30639$, $\mathcal{K}_1 = \{1, 5\}$.

Define Case II as a system with same settings as in the simulations for Figure 1(b), except for the following parameters:

- $\mu_1 = 7.74574$, $\epsilon_1^0 = 0.77457$, $\epsilon_1 = 7.74574$;
- $\mu_2 = 7.46818$, $\epsilon_2^0 = 1.77913$, $\epsilon_2 = 8.89565$;
- $\mu_3 = 6.32019$, $\epsilon_3^0 = 2.65737$, $\epsilon_3 = 8.85791$;
- $\mu_4 = 4.66817$, $\epsilon_4^0 = 3.27786$, $\epsilon_4 = 8.19465$;
- $\mu_5 = 4.62779$, $\epsilon_5^0 = 4.82040$, $\epsilon_5 = 9.64079$;
- $\lambda_1 = 11.04970$, $\mathcal{K}_1 = \{2, 3, 5\}$;
- $\lambda_2 = 8.27302$, $\mathcal{K}_2 = \{2, 3\}$;
- $\lambda_3 = 11.04970$, $\mathcal{K}_3 = \{2, 3, 5\}$.

Note that both examples are taking instances of the randomly generated system in simulations in Sections VII-A1 and VII-A2.

APPENDIX D

SETTINGS FOR SIMULATIONS IN SECTION VII-B

Consider ten server groups with the following parameters:

- $\mu_1 = 2.42523$, $\epsilon_1^0 = 0.06548$, $\epsilon_1 = 0.242523$;
- $\mu_2 = 2.41588$, $\epsilon_2^0 = 0.04974$, $\epsilon_2 = 0.207254$;
- $\mu_3 = 2.38966$, $\epsilon_3^0 = 0.04279$, $\epsilon_3 = 0.20377$;
- $\mu_4 = 2.22434$, $\epsilon_4^0 = 0.02630$, $\epsilon_4 = 0.14613$;
- $\mu_5 = 1.75822$, $\epsilon_5^0 = 0.07228$, $\epsilon_5 = 0.60241$;
- $\mathcal{K}_1 = \{1, 5, 6, 10\}$, $B_1 = 20$;
- $\mathcal{K}_2 = \{1, 2, 3, 4, 5, 7, 8, 9\}$, $B_2 = 20$;
- $\mathcal{K}_3 = \{1, 6, 7, 10\}$, $B_3 = 20$; and $\mathcal{K}_4 = \{2\}$, $B_4 = 20$.

Note that all the numbers are generated by a pseudo-random number generator, and the unit of service rates μ_k is $10^{-4} s^{-1}$: the number of processed jobs per second. The values of μ_k are normalized to be sufficiently small that we can observe a positive number of blocked jobs in Figure 6, and the heavy traffic condition can be achieved during the peak hours as presented in Section VII-B.

ACKNOWLEDGMENT

Jing Fu's research is supported by the Australian Research Council (ARC) Centre of Excellence for the Mathematical and Statistical Frontiers (ACEMS) and ARC Laureate Fellowship FL130100039.

REFERENCES

- [1] Emerson Network Power, "State of the data center 2011," 2011. [Online]. Available: <http://www.emersonnetworkpower.com/en-US/Solutions/infographics/Pages/2011DataCenterState.aspx>
- [2] Natural Resources Defense Council, "Americas data centers consuming massive and growing amounts of electricity," Aug. 2014. [Online]. Available: <http://www.nrdc.org/media/2014/140826.asp>
- [3] D. Kliazovich, P. Bouvry, F. Granelli, and N. L. S. da Fonseca, "Energy consumption optimization in cloud data centers," in *Cloud Services, Networking, and Management*, N. L. S. da Fonseca and R. Boutaba, Eds. John Wiley & Sons, Inc, Apr. 2015, pp. 191–215. [Online]. Available: <http://dx.doi.org/10.1002/9781119042655.ch8>
- [4] S. Albers, F. Müller, and S. Schmelzer, "Speed scaling on parallel processors," *Algorithmica*, vol. 68, no. 2, pp. 404–425, Feb. 2014.
- [5] J. Lenhardt, K. Chen, and W. Schiffmann, "Energy-efficient web server load balancing," *IEEE Syst. J.*, vol. 11, no. 2, pp. 878–888, Jun. 2017.
- [6] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1378–1391, Oct. 2013.
- [7] Y. Yao, L. Huang, A. B. Sharma, L. Golubchik, and M. J. Neely, "Power cost reduction in distributed data centers: A two-time-scale approach for delay tolerant workloads," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 200–211, Jan. 2014.
- [8] E. Hyttiä, R. Righter, and S. Aalto, "Task assignment in a heterogeneous server farm with switching delays and general energy-aware cost structure," *Performance Evaluation*, vol. 75–76, pp. 17–35, 2014.
- [9] M. E. Gebrehiwot, S. Aalto, and P. Lassila, "Near-optimal policies for energy-aware task assignment in server farms," in *Proc. CCGrid 2017*. Madrid, Spain: IEEE Press, May 2017, pp. 1017–1026.
- [10] T. Lin, T. Alpcan, and K. Hinton, "A game-theoretic analysis of energy efficiency and performance for cloud computing in communication networks," *IEEE Syst. J.*, vol. 11, no. 2, pp. 649–660, Jun. 2017.
- [11] W. Shi, C. Wu, and Z. Li, "An online auction mechanism for dynamic virtual cluster provisioning in geo-distributed clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 3, pp. 677–688, Mar. 2017.
- [12] W. Q. M. Guo, A. Wadhawan, L. Huang, and J. T. Dudziak, "Server farm management," Patent 8,626,897, Jan., 2014.
- [13] A. Hameed, A. Khoshkbarfoushha, R. Ranjan, P. P. Jayaraman, J. Kolodziej, P. Balaji, S. Zeadally, Q. M. Malluhi, N. Tziritas, A. Vishnu, S. U. Khan, and A. Zomaya, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," *Computing*, vol. 98, no. 7, pp. 751–774, Jun. 2016.
- [14] A. Gandhi and M. Harchol-Balter, "How data center size impacts the effectiveness of dynamic power management," in *Proc. 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Monticello, IL, USA: IEEE, Sep. 2011, pp. 1164–1169.
- [15] T. Lu, M. Chen, and L. L. H. Andrew, "Simple and effective dynamic provisioning for power-proportional data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1161–1171, Apr. 2013.
- [16] Z. Rosberg, Y. Peng, J. Fu, J. Guo, E. W. M. Wong, and M. Zukerman, "Insensitive job assignment with throughput and energy criteria for processor-sharing server farms," *IEEE/ACM Trans. Netw.*, vol. 22, no. 4, pp. 1257–1270, Aug. 2014.
- [17] J. Fu, J. Guo, E. W. M. Wong, and M. Zukerman, "Energy-efficient heuristics for insensitive job assignment in processor-sharing server farms," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2878–2891, Dec. 2015.
- [18] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.
- [19] M. Chowdhury, M. R. Rahman, and R. Boutaba, "Vineyard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, Feb. 2012.
- [20] F. Esposito, D. Di Paola, and I. Matta, "On distributed virtual network embedding with guarantees," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 569–582, Feb. 2016.
- [21] J. Fu, B. Moran, J. Guo, E. W. M. Wong, and M. Zukerman, "Asymptotically optimal job assignment for energy-efficient processor-sharing server farms," *IEEE J. Sel. Areas Commun.*, 2016.
- [22] U. Ayesta, M. Erausquin, M. Jonckheere, and I. M. Verloop, "Scheduling in a random environment: stability and asymptotic optimality," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 258–271, Feb. 2013.
- [23] I. M. Verloop, "Asymptotically optimal priority policies for indexable and non-indexable restless bandits," *Ann. Appl. Probab.*, vol. 26, no. 4, pp. 1947–1995, Aug. 2016.
- [24] V. Gupta, "Stochastic models and analysis for resource management in server farms," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 2011.
- [25] Y. Sakuma, "Asymptotic behavior for MAP/PH/c queue with shortest queue discipline and jockeying," *Oper. Res. Lett.*, vol. 38, no. 1, pp. 7–10, Jan. 2010.
- [26] P. Whittle, "Restless bandits: Activity allocation in a changing world," *J. Appl. Probab.*, vol. 25, pp. 287–298, 1988.
- [27] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queuing network control," *Math. Oper. Res.*, vol. 24, no. 2, pp. 293–305, May 1999.
- [28] R. J. Weber and G. Weiss, "On an index policy for restless bandits," *J. Appl. Probab.*, no. 3, pp. 637–648, Sep. 1990.
- [29] J. Niño-Mora, "Restless bandits, partial conservation laws and indexability," *Advances in Applied Probability*, vol. 33, no. 1, pp. 76–98, 2001.
- [30] J. Niño-Mora, "Dynamic allocation indices for restless projects and queueing admission control: a polyhedral approach," *Mathematical programming*, vol. 93, no. 3, pp. 361–413, 2002.
- [31] J. Niño-Mora, "Dynamic priority allocation via restless bandit marginal productivity indices," *TOP*, vol. 15, no. 2, pp. 161–198, Sep. 2007.
- [32] X. Wei and M. J. Neely, "Data center server provision: Distributed asynchronous control for coupled renewal systems," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2180 – 2194, Aug. 2017.
- [33] M. Pore, Z. Abbasi, S. K. S. Gupta, and G. Varsamopoulos, "Techniques to achieve energy proportionality in data centers: A survey," in *Handbook on Data Centers*. Springer, Mar. 2015, pp. 109–162.
- [34] K. Li, "Quantitative modeling and analytical calculation of elasticity in cloud computing," *IEEE Transactions on Cloud Computing*, 2017.
- [35] S. Sebastiao, M. Amoretti, A. L. Lafuente, and A. Scala, "A holistic approach for collaborative workload execution in volunteer clouds," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 28, no. 2, p. 14, 2018.
- [36] M. Guo, Q. Guan, and W. Ke, "Optimal scheduling of VMs in queueing cloud computing systems with a heterogeneous workload," *IEEE Access*, vol. 6, pp. 15 178–15 191, 2018.
- [37] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proc. the Third ACM Symposium on Cloud Computing*. ACM, 2012, p. 7.
- [38] J. Fu, B. Moran, and P. G. Taylor, "Restless bandits in action: Resource allocation, competition and reservation," *arXiv: 1804.02100*, Apr. 2018. [Online]. Available: <https://arxiv.org/abs/1804.02100>
- [39] J. Wilkes, "More Google cluster data," Google research blog, Nov. 2011, posted at <http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html>, accessed at Jul. 8, 2019.
- [40] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format + schema," Google Inc., Mountain View, CA, USA, Technical Report, Nov. 2011, revised 2014-11-17 for version 2.1. Posted at <https://github.com/google/cluster-data>, accessed at Jul. 8, 2019.
- [41] W. Winston, "Optimality of the shortest line discipline," *J. Appl. Probab.*, vol. 14, no. 1, pp. 181–189, Mar. 1977.
- [42] V. Gupta, M. Harchol-Balter, K. Sigman, and W. Whitt, "Analysis of join-the-shortest-queue routing for web server farms," *Perform. Eval.*, vol. 64, no. 9-12, pp. 1062–1081, Oct. 2007.
- [43] E. Altman, U. Ayesta, and B. J. Prabhu, "Load balancing in processor sharing systems," *Telecommun. Syst.*, vol. 47, no. 1-2, pp. 35–48, Jun. 2011.
- [44] N. Bansal and M. Harchol-Balter, "Analysis of SRPT scheduling: Investigating unfairness," in *ACM SIGMETRICS 2001*. Massachusetts, USA: ACM, Jun. 2001, pp. 279–290.
- [45] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.
- [46] S. M. Ross, *Applied probability models with optimization applications*. Dover Publications (New York), 1992.
- [47] M. I. Freidlin and A. D. Wentzell, *Random perturbations of dynamical systems*. Springer Science & Business Media, 2012, translated by J. Szücs.