Asymptotically Optimal Job Assignment for Energy-Efficient Processor-Sharing Server Farms

Jing Fu, *Member, IEEE,* Bill Moran, *Member, IEEE,* Jun Guo, *Member, IEEE,*
Eric W. M. Wong, *Senior Member, IEEE,* and Moshe Zukerman, *Fellow, IEEE*

***Abstract*—We study the problem of job assignment in a large-scale realistically-dimensioned server farm comprising multiple processor-sharing servers with different service rates, energy consumption rates, and buffer sizes. Our aim is to optimize the energy efficiency of such a server farm by effectively controlling carried load on networked servers. To this end, we propose a job assignment policy, called Most energy-efficient available server first Accounting for Idle Power (MAIP), which is both scalable and near optimal. MAIP focuses on reducing the productive power used to support the processing service rate. Using the framework of semi-Markov decision process we show that, with exponentially distributed job sizes, MAIP is equivalent to the well-known Whittle's index policy. This equivalence and the methodology of Weber and Weiss enable us to prove that, in server farms where a loss of jobs happens if and only if all buffers are full, MAIP is asymptotically optimal as the number of servers tends to infinity under certain conditions associated with the large number of servers as we have in a real server farm. Through extensive numerical simulations, we demonstrate the effectiveness of MAIP and its robustness to different job-size distributions, and observe that significant improvement in energy efficiency can be achieved by utilizing knowledge of energy consumption rate of idle servers.**

***Index Terms*—Energy efficiency, job assignment, bandit problem, processor sharing, server farm.**

I. Introduction

THE data-center industry, with over 500 thousand data centers worldwide [1], has been growing in parallel with the dramatic increase in global Internet traffic. An estimated 91 billion kWh of electricity was consumed by U.S. data centers in 2013, and this consumption rate continues to grow, resulting in $13 billion annually for electricity bills and potentially nearly 100 million metric tons of carbon pollution per year by 2020 [2]. Servers account for the major portion of energy consumption of data centers [3]. Our aim here is to describe optimal scheduling/dispatching strategies for incoming job requests in a server farm so as to improve energy efficiency.

This has been a topic of interest for some time, with approaches such as speed scaling to reduce energy consumption by controlling server speed(s) [4]–[8]. Right-sizing of server farms has been done by powering servers on/off according to traffic load [9]–[12], and by switching servers between active/sleep mode according to number of waiting jobs [13]. In [14], [15] resource allocation methodologies are used to distribute the power budget for energy conservation.

Rapid improvements in computer hardware have resulted in frequent upgrades of parts of the server farms. This, in turn, has led to server farms with a range of different computer resources (heterogeneous servers) being deployed [16]. Such heterogeneity significantly complicates optimization, since each server needs to be considered individually. Despite the complexity, we are able to improve here the energy efficiency of heterogeneous server farm via appropriate scalable job assignment policies that are applicable to server farms with tens of thousands of servers. For the purposes of this paper, a server farm is postulated to have a fixed number of servers with no possibility of powering off during the time period under consideration; this, in practice, could apply to periods during which no powering off takes place. In this way, the job assignment policies described here can be combined with the right-sizing techniques mentioned above, as appropriate. Note that frequent powering off/on increases wear and tear and the need for costly replacement and maintenance [17].

Here, we consider a system in which idle servers may have non-negligible energy consumption rate [18]. In [6], [19], [20], job assignment policies have been discussed without consideration of idle power (energy consumption rate of idle servers), and in [9], [11], [21], such policies have been considered for a server pool with identical servers. In the server farms considered, servers are not assumed to be identical. To the best of our knowledge, there is no published work that considers idle power in designing job assignment methodologies for a given heterogeneous server pool.

Other job assignment policies in the literature (e.g., [22]–[24]) have considered scenarios with infinite buffer size and have sought to minimize delay. We consider a server farm with parallel finite-buffer queues, and with heterogeneous server speeds, energy consumption rates, and buffer sizes. As in [19], the energy efficiency of a server farm is defined as the ratio of the *long-run expected job departure rate* divided by the expected energy consumption rate. It forms the objective function of our optimization strategy. This objective function represents the amount of useful work (e.g., data rate, throughput, processes per second) per watt.

The processor sharing (PS) discipline is imposed on each queue, so that all jobs on the same queue share the processing

This work was supported by CityU Strategic Research Grants Project No. 7004434 and Project No. 7004611. Most of the work was done when J. Fu and J. Guo were with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong.

J. Fu is with the School of Mathematics and Statistics, The University of Melbourne, Melbourne, Australia (e-mail: jing.fu@unimelb.edu.au).

B. Moran is with the School of Engineering, RMIT University, Melbourne, Australia (e-mail:bill.moran@rmit.edu.au).

J. Guo is with the College of Computer Science and Technology, Dongguan University of Technology, Dongguan, China (e-mail: guojun@dgut.edu.cn).

E. W. M. Wong and M. Zukerman are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong (e-mail: eeewong@cityu.edu.hk; m.zu@cityu.edu.hk).

capacity and are served at the same rate. The PS discipline avoids unfair delays for those jobs that are preceded by extremely large jobs, making it an appropriate model for web server farms [25], [26], where job-size distributions are highly variable [27], [28]. The finite buffer size queuing model with PS discipline can be applied in situations where a minimum service rate is required for processing a job in the system [29]. In communication systems, broader applications of PS queues have been studied; e.g., [30], [31].

A key feature of our approach is to model the problem as an instance of the Restless Multi-Armed Bandit Problem (RMABP) [32] in which, at each epoch, a policy chooses a server to be *tagged* for a new job assignment (other servers are said to be *untagged*). The general RMABP has been proved SPACE-hard [33], and this has led to studies in scalable and near-optimal approximations, such as *index policies*. An index policy selects a set of tagged servers at any epoch according to their state-dependent indices.

We consider a large-scale realistically-dimensioned server farm that cannot reject a job if it has buffer space available. Such a situation would occur, for instance, where a server farm owner is unable to replace all older servers simultaneously, and so legacy inefficient servers are needed to meet a service level agreement. Buffering spill-over creates dependencies between servers, and requires us to postulate *uncontrollable states* [34]. In other words, the constraint on the number of tagged servers in conventional RMABP has to be replaced by a constraint on the number of tagged servers in controllable states. As far as we are aware, there are no theoretical results on the asymptotic optimality of an index policy for a multi-server system with finite buffer size, where loss of jobs happens if and only if all buffers are full. A further discussion on existing related work is provided in Section II.

The contribution of this paper is listed as follows.

- We propose a new job assignment method to attempt maximization of energy efficiency. The new proposed policy is referred to as the *Most energy-efficient available server first Accounting for Idle Power* (MAIP). MAIP prioritizes the most energy-efficient servers that are available (that is, servers with at least one vacant slot in their buffers) and requires the energy consumption rate of idle servers as an input parameter for decision making. This policy provides a model of a real system with significant energy consumption rate in idle states. MAIP is scalable and requires only binary state information of servers, making it suitable for an environment with frequently changing server states. Our server farm model is centralized and is applicable to a local system with frequently changing information, which for our case is the binary information of server states. We note that Google has built a centralized control mechanism for network routing and management that monitors all link states and is scalable for Google's building-scale data center [35].
- We prove, remarkably, that when job sizes are exponentially distributed, the Whittle's index policy is equivalent to MAIP, and that it is asymptotically optimal for our server farm comprised of multiple groups of identical servers as the numbers of servers in these groups tend to infinity. It is reasonable to assume that if the total number of servers in a server farm is very large, then the number of servers bought in a single batch, or over a short period of time during which the technology is not improving, will also be large. In any case, there is cost benefit in buying in bulk, so that the number of servers purchased at once is likely to be large. More importantly, the typical lifespan of a server is in the range of 3 years [36], [37]. Accordingly, a modern server farm is likely to be categorized into several server groups, each of which contains a large number of servers of the same or similar type and attributes that were bought at the same time, or over a short time period.
  The well-known Whittle's index relaxation enables decomposition of a complex RMABP problem into multiple sub-problems, assumed computationally feasible [32]. Note that, in the general case, Whittle's index does not necessarily exist, and even if it does, a closed form solution is often unavailable. As mentioned before and in Section II, the buffer constraint in our case enforces the need for uncontrollable states and, in turn, prevents direct application of previous asymptotic optimality results on RMABP to our problem.
- We demonstrate numerically the effectiveness of MAIP by comparing it with a baseline policy that prioritizes the most energy-efficient available serves but ignores idle power. Although, as mentioned above, powering off servers is not considered here, the performance of the baseline policy can be significantly improved by taking the idle power into consideration (MAIP) in terms of energy efficiency. MAIP is demonstrated numerically to be almost insensitive to the shape of job-size distributions. We also demonstrate the applicability of MAIP for a large server farm with significant cost of job reassignment.

The remainder of this paper is organized as follows. In Section II, we discuss the related work on job assignment policies. In Section III, we describe the server farm model. In Section IV, we propose the MAIP policy, and in Section V, we give the proof for the asymptotic optimality of MAIP. We present numerical results in Section VI and conclusions are given in Section VII.

II. Related Work

Queueing models associated with job assignment among multiple servers with and without jockeying (reassignment actions of incomplete jobs) have been studied since 1958 [38]. Most existing work has focused on job assignment policies that aim to improve the system performance under a first-come-first-served (FCFS) discipline such as *Join-the-Shortest-Queue* (JSQ).

For the non-jockeying case, JSQ under PS has been analyzed in [22], [23], [25], [39]. Bonomi [22] proved optimality of JSQ for the processor sharing case under a general arrival process, a Markov departure process, and homogeneous servers while, for a non-exponential job-size distribution, a counter-example to optimality of JSQ has been given by Whitt [39]. Gupta [23] provided an analysis for the approximation of

the state distribution of a system under JSQ with homogeneous PS servers and general job-size distributions. Gupta also showed the optimality of JSQ in terms of average delay for a system comprising servers with two different service rates.

Server farm applications of JSQ with jockeying policies for FCFS have been studied in [24], [40], [41]. In these papers, when the difference between the longest and shortest queue sizes achieves a threshold, a jockeying action is triggered. Different values of the threshold clearly result in different JSQ policies. These publications focus on the calculation of the equilibrium distribution of the lengths of queues.

Energy efficiency for a multi-queue heterogeneous system with infinite buffers and set-up delay has been studied in [42], where the authors assume zero energy consumption rate when a server is idle. Hyytiä *et al.* [42] have shown that the M/G/1-LCFS is insensitive to the shape of the distribution of set-up delay while this insensitivity is lost in the M/G/1-PS. In [43], energy-efficient job assignment in a system with heterogeneous servers and homogeneous job has been considered, where jobs were queued in an infinite public buffer without waiting room on each server and no cost were consumed for an idle server. The authors analyzed the coinciding of individual optimality (minimizing the cost of one job) and social optimality (minimize the sum of the cost of all jobs) that were, both proved in this paper, threshold style in certain situations.

Energy-aware PS multi-queue systems with jockeying have been studied in [19], [20], where the optimization problem is characterized as a semi-Markov decision process (SMDP) [44]. In that paper, maximization of the ratio of job throughput to power consumption (the ratio of the long-run average job departure rate to the long-run average energy consumption rate) is introduced as a measure of performance. The use of long-run average reward per unit cost (e.g., time consumption, energy consumption, etc.) as an objective function in [19] generalizes long-run average service quality per unit time, studied previously.

In this paper, we consider a large-scale realistically-dimensioned server farm model with heterogeneous servers. The jockeying case discussed in [19], [20] is more appropriate for a localized server farm, in which the cost of jockeying actions is negligible. For a server farm with significant jockeying costs, a simple scalable job assignment policy without jockeying is more attractive. A similar dynamic programming methodology in which the computational complexity increases linearly in the number of states can be applied to our case. Unfortunately, in the non-jockeying server farm the number of states increases exponentially in the number of servers, so that the optimal solution is limited to very small cases with a few servers.

The asymptotic optimality analysis in [45]–[49] is applicable for job assignment policies in a multi-queue system with infinite buffer size on each queue. In particular, Weber and Weiss [45] proved, under certain conditions, the asymptotic optimality of Whittle's index policies, as conjectured by Whittle [32]. Mandelbaum and Stolyar [46] considered a similar model in continuous time, and proved that a simple generalized rule, called the cμ-rule, asymptotically minimizes instantaneous and cumulative holding costs in a queueing system with multiple-parallel flexible servers and multi-class jobs when the system has heavy traffic, and a stability condition is satisfied. Also, the holding cost rate in [46] is assumed to be increasing and convex while, in our problem, the holding cost rate is not necessarily even increasing in the workload of a server. Nazarathy and Weiss [47] proposed a method for the control problem of a multi-server queueing network over a finite time horizon. They proved that the method is asymptotically optimal in the minimization of the total inventory cost over a finite time horizon. Ayesta *et al.* [48] have studied a preemptive queue with infinite buffer and multiple users as a model for the flow-level behavior of end-users in a narrowband HDR wireless channel (CDMA 1xEV-DO). They discussed conditions for the stability and the asymptotic optimality of policies under which users (channels) are selected. Verloop [49] extended the proof of asymptotic optimality for the Whittle's index in [45] to cases with several classes of bandits, with arrivals of new bandits, and with multiple actions per bandit.

Other publications on asymptotic optimality in job assignment include [50], [51]. Atar and Shifrin [50] analyzed a G/G/1 queue with finite buffer and multiple classes of arriving jobs, where all jobs share the finite buffer capacity of the queue. They also prove asymptotic optimality of their method under some buffer size constraints, and under a throughput time (delay) constraint in the presence of further restrictions. Larrañaga *et al.* [51] analyzed a system with multiple users (modeled as multiple bandits), aiming at minimizing the average cost: a combination of convex (also non-decreasing) holding costs and user impatience. This work contains uncontrollable states as a bandit cannot be played when the number of corresponding users is zero, but the non-decreasing holding cost constraint, which simplifies the asymptotic optimality argument, cannot be guaranteed in our problem.

This large, but by no means complete, collection of related work, contains no asymptotic optimality result directly applicable to our problem of a multi-server queue system with a buffer constraint on each queue, requiring the presence of uncontrollable states as mentioned in Section I. As far as we are aware, there is no result on the asymptotic optimality of an index policy in this context.

III. Model

We consider a heterogeneous server farm modeled as a multi-queue system where reassignment of incomplete jobs is not allowed. For the reader's convenience, Table I provides a list of symbols that are frequently used in this paper.

The server farm has a total of $K \geq 2$ servers, forming the set $\mathcal{K} = \{1, 2, \ldots, K\}$. These servers are characterized by their service rates, energy consumption rates, and buffer sizes. For $j \in \mathcal{K}$, we denote by μ_j the service rate of server j and by B_j its buffer size. The energy consumption rate of server j is ε_j when it is busy and ε_j^0 when it is idle, respectively, where $\varepsilon_j > \varepsilon_j^0 \geq 0$. We refer to the ratio $\mu_j/(\varepsilon_j - \varepsilon_j^0)$ as the *effective energy efficiency* of server j. Note that this definition of energy efficiency for each server in the system that takes

TABLE I
SUMMARY OF FREQUENTLY USED SYMBOLS

Symbol	Definition
$\mathcal{K}$	Set of servers in the system
K	Number of servers in the system
B_j	Buffer size of server j
μ_j	Service rate of server j
ε_j	Energy consumption rate of server j when it is busy
ε_j^0	Energy consumption rate of server j when it is idle
$\mu_j/(\varepsilon_j - \varepsilon_j^0)$	Effective energy efficiency of server j
λ	Job arrival rate
$\mathcal{L}^\phi$	Job throughput of the system under policy ϕ
$\mathcal{E}^\phi$	Power consumption of the system under policy ϕ
$\mathcal{L}^\phi/\mathcal{E}^\phi$	Energy efficiency of the system under policy ϕ

into account the effect of idle power is key to the design of the MAIP policy proposed in this paper.

Job arrivals follow a Poisson process with rate λ, indicating the average number of arrivals per time unit. An arriving job is assigned to one of the servers with at least one vacant slot in its buffer, subject to the control of an assignment policy ϕ. If all buffers are full, the arriving job is lost.

We assume that job sizes (in units) are independent and identically distributed, and normalize without loss of generality the average size of jobs to one. Each server j serves its jobs at a total rate of μ_j using the PS service discipline.

Our considerations are limited to realistic cases, and assume that the ratio of the arrival rate to the total service rate, i.e., $\rho \stackrel{\text{def}}{=} \lambda/\sum_{j=1}^{K}\mu_j$, is sufficiently large to be economically justifiable but not too large to violate the required quality of service. We refer to ρ as the *normalized offered traffic*.

The job throughput of the system under policy ϕ, which is equivalent to the long-run average job departure rate, is denoted by $\mathcal{L}^\phi$. The power consumption of the system under policy ϕ, which is equivalent to the long-run average energy consumption rate, is denoted by $\mathcal{E}^\phi$. By definition, $\mathcal{L}^\phi/\mathcal{E}^\phi$ is the energy efficiency of the system under policy ϕ.

IV. JOB ASSIGNMENT POLICY: MAIP

Here we provide details of the MAIP policy. Note that, with a non-jockeying policy, the server farm scheduler makes assignment decisions only at arrival events and assigns a new job to one of the available servers in the system. MAIP is designed in such a way that takes into account the effect of idle power. The key idea of MAIP can be conveniently explained by using a simple example.

Consider a system with two servers only, where $\mu_1 = \mu_2 = 1$, $\varepsilon_1 = 2$, $\varepsilon_1^0 = 1$, $\varepsilon_2 = 2.5$, and $\varepsilon_2^0 = 2$. It is clear that in this example $\varepsilon_1 < \varepsilon_2$ and $\varepsilon_1^0 < \varepsilon_2^0$. If a job arrives when both servers are idle, the scheduler has two choices:

1) Assigning the job to server 1 makes server 1 busy. The energy consumption rate of the whole system becomes $\varepsilon_1 + \varepsilon_2^0 = 4$.
2) Assigning the job to server 2 makes server 2 busy. The energy consumption rate of the whole system becomes instead $\varepsilon_2 + \varepsilon_1^0 = 3.5$.

Since $(\varepsilon_1 + \varepsilon_2^0) > (\varepsilon_2 + \varepsilon_1^0)$, which is equivalently $(\varepsilon_1 - \varepsilon_1^0) > (\varepsilon_2 - \varepsilon_2^0)$, and since both servers have the same service rate, choosing server 2 for serving the job in this particular example turns out to be better in terms of the energy efficiency of the system, despite the fact that server 2 consumes more power when busy than server 1 does.

Intuitively, in the situation where power consumption of idle servers in a system is not necessarily negligible, the energy used by the system can be categorized into two parts: *productive* and *unproductive*. The productive part contributes to job throughput, whereas the unproductive part is a waste of energy. For a server j, when it is idle, the service rate is 0 accompanied by an energy consumption rate of ε_j^0; when it is busy, the service rate becomes μ_j and the energy consumption rate increases to ε_j. We regard the additional service rate $\mu_j - 0$ as a reward at the cost of an additional energy consumption rate $\varepsilon_j - \varepsilon_j^0$. In other words, if jobs are assigned to server j, the productive power used to support the service rate μ_j is effectively $\varepsilon_j - \varepsilon_j^0$. Accordingly, productive power is our main concern in the design of MAIP.

Since MAIP aims for energy-efficient job assignment, for convenience of description, we label the servers according to their effective energy efficiency. In particular, in the context of MAIP, server i is defined to be more energy-efficient than server j if and only if $\mu_i/(\varepsilon_i - \varepsilon_i^0) > \mu_j/(\varepsilon_j - \varepsilon_j^0)$. That is, for any pair of servers i and j, if $i < j$, we have $\mu_i/(\varepsilon_i - \varepsilon_i^0) \geq \mu_j/(\varepsilon_j - \varepsilon_j^0)$. Then, MAIP works by always selecting a server with the highest effective energy efficiency among all servers that contain at least one vacant slot in their buffers, where ties are broken arbitrarily. As a result of this design, MAIP is a simple approach that requires only binary state information (i.e., available or unavailable) from each server for its implementation.

V. ANALYSIS

Here we give a precise definition of our optimization problem as described informally in Section V-A. In Section V-B, the Whittle's index policy for our problem is described, and in Section V-C, with exponentially distributed job sizes, we prove the indexability of our server farm model and present a closed-form expression of Whittle's index, producing the equivalence of Whittle's index policy and MAIP. In Section V-D, the proof of asymptotic optimality of Whittle's index policy is presented, leading to the asymptotic optimality of MAIP.

As mentioned in Section I, MAIP is asymptotically optimal for our server farm comprised of multiple groups of identical servers as the numbers of servers in these groups tend to infinity, which is appropriate for a modern server farm that always buys and upgrades a large number of servers of the same type and attributes at the same time.

A. Stochastic Process

Let $\mathcal{N}_j$ denote the set of all states of server j, where the state, n_j is the number of jobs queueing or being served at server j. Thus, $\mathcal{N}_j = \{0, 1, \ldots, B_j\}$, where $B_j \geq 2$ is the buffer size for server j. For server j, states $0, 1, \ldots, B_j - 1$ are called *controllable*, and the state B_j is called *uncontrollable*.

The set of controllable states for server j, in which the server is available to be tagged, is denoted by $\mathcal{N}_j^{\{0,1\}} = \{0, 1, \ldots, B_j - 1\}$ while, for the uncontrollable state in the set $\mathcal{N}_j^{\{0\}} = \{B_j\}$, the server is forced to be untagged because it cannot accept jobs. Recall that a tagged server has been defined in Section IV as a server selected to accept new jobs. The vector $\mathbf{n} = (n_1, n_2, \ldots, n_K)$ represents the state of the multi-queue system, $n_j \in \mathcal{N}_j$, $j \in \mathcal{K}$. The set of all such states $\mathbf{n}$ is denoted by $\mathcal{N}$, the sets of uncontrollable and controllable states in $\mathcal{N}$ are, respectively,

$$\begin{aligned} \mathcal{N}^{\{0\}} &= \left\{ \mathbf{n} \in \mathcal{N} \mid n_j \in \mathcal{N}_j^{\{0\}},\ \forall j \in \mathcal{K} \right\}, \\ \mathcal{N}^{\{0,1\}} &= \left\{ \mathbf{n} \in \mathcal{N} \mid \mathbf{n} \notin \mathcal{N}^{\{0\}} \right\}. \end{aligned} \tag{1}$$

We define $\mathbf{X}^{\phi}(t) = (X_1^{\phi}(t), X_2^{\phi}(t), \ldots, X_K^{\phi}(t))$ to be a vector of random variables representing the state at time t under policy ϕ of the stochastic process of the multi-queue system. We set without loss of generality the initial state $\mathbf{X}^{\phi}(0) = \mathbf{x}(0)$, $\mathbf{x}(0) \in \mathcal{N}$.

Decisions made on job arrivals rely on the values of $\mathbf{X}(t)$ just before an arrival occurs. We use $a_j^{\phi}(t)$, $j \in \mathcal{K}$ as an indicator of activity at time t under policy ϕ, so that $a_j^{\phi}(t) = 1$ if server j is tagged, and $a_j^{\phi}(t) = 0$, otherwise. Then $\sum_{j=1}^{K} a_j^{\phi}(t) \leq 1$ for all $t > 0$. All job assignment policies considered are *stationary*, so that we also use $a_j^{\phi}(\mathbf{n})$, $\mathbf{n} \in \mathcal{N}$, to represent the action we take on the stochastic process when the system is in state $\mathbf{n}$. A policy ϕ comprises those $\mathbf{a}^{\phi}(\mathbf{n}) = (a_1^{\phi}(\mathbf{n}), a_2^{\phi}(\mathbf{n}), \ldots, a_K^{\phi}(\mathbf{n}))$, for all $\mathbf{n} \in \mathcal{N}$.

For clarity of presentation, we define a mapping $R_j : \mathcal{N}_j \to R$, where $R_j(n_j)$ ($n_j \in \mathcal{N}_j$) is the reward rate of server j in state n_j. Let $\mathcal{R}_j$ be the set of all such mappings R_j. Then, for a given vector of mappings $\mathbf{R} = (R_1, R_2, \ldots, R_K)$, we define the *long-run average reward* under policy ϕ to be

$$\gamma^{\phi}(\mathbf{R}) = \lim_{t \to +\infty} \frac{1}{t} \mathbb{E} \left\{ \int_0^t \sum_{j \in \mathcal{K}} R_j(X_j^{\phi}(u)) du \right\}. \tag{2}$$

We refer to $\mathbf{R}$ as the *reward rate function*. Along similar lines, we consider $\mu_j(n_j)$ and $\varepsilon_j(n_j)$, the service rate and energy consumption rate of server j in state n_j, respectively, as rewards; that is, $\mu_j, \varepsilon_j \in \mathcal{R}_j$. As defined in Section III, $\mu_j(n_j) = \mu_j$, $\varepsilon_j(n_j) = \varepsilon_j$ for $n_j > 0$, $\mu_j(0) = 0$ and $\varepsilon_j(0) = \varepsilon_j^0$, where $\mu_j > 0$, $\varepsilon_j > \varepsilon_j^0 \geq 0$, $j \in \mathcal{K}$. For the vectors $\mu = (\mu_1, \mu_2, \ldots, \mu_K)$ and $\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_K)$, the long-run average job service rate of the entire system is, then, $\gamma^{\phi}(\mu)$, and the long-run average energy consumption rate of the system is $\gamma^{\phi}(\varepsilon)$. The problem of maximizing energy efficiency is then encapsulated in

$$\max_{\phi} \frac{\gamma^{\phi}(\mu)}{\gamma^{\phi}(\varepsilon)}. \tag{3}$$

Based on the definition given above, we formally define MAIP as follows.

$$a_j^{\mathrm{MAIP}}(\mathbf{n}) = \begin{cases} 1, & \mathbf{n} \in \mathcal{N}^{\{0,1\}}, \\ & j = \min \operatorname{argmax}_{j \in \mathcal{K}: n_j \in \mathcal{N}_j^{\{0,1\}}} \frac{\mu_j}{\varepsilon_j - \varepsilon_j^0}, \\ 0, & otherwise. \end{cases} \tag{4}$$

As mentioned before, to show the asymptotic optimality of MAIP in Section V-D, we will introduce necessary knowledge about the *Whittle's index* in Sections V-B and will obtain its equivalence to MAIP in Section V-C.

B. Whittle's Index

In 1979, Gittins [52] produced the optimal solution for the general multi-armed bandit problem (MABP); the so-called *Gittins' index* policy. Relaxing the constraint that only one machine (project/bandit/process) is played at a time, and only the played machine changes state, Whittle [32] published a more general model, the *restless multi-armed bandit problem* (RMABP) and proposed as an index the so-called *Whittle's index* as an approximation for optimality.

The general definition of Whittle's index for our problem is given here; a closed-form expression will be provided in Section V-C for the case when job sizes are exponentially distributed.

According to [19, Theorem 1], there exists a value $e^* > 0$, given by

$$e^* = \max_{\phi} \left\{ \frac{\gamma^{\phi}(\mu)}{\gamma^{\phi}(\varepsilon)} \right\}, \tag{5}$$

so that our optimization problem (3) can be written as

$$\sup_{\phi} \left\{ \gamma^{\phi}(\mathbf{R}) : \sum_{\substack{j \in \{1,2,\ldots,K\}: \\ X_j^{\phi}(t) \in \mathcal{N}_j^{\{0,1\}}}} a_j^{\phi}(t) = 1,\ \forall t \geq 0 \right\} \tag{6}$$

where the reward rate function $\mathbf{R} = (R_1, R_2, \ldots, R_K)$, $R_j \in \mathcal{R}_j$, $R_j(n_j) = \mu_j(n_j) - e^* \varepsilon_j(n_j)$, $j \in \mathcal{K}$.

Following the Whittle's index approach, we relax our problem (6) as

$$\begin{aligned} &\sup_{\phi} \lim_{t \to +\infty} \frac{1}{t} \mathbb{E} \left\{ \int_0^t \sum_{j \in \mathcal{K}} R_j(X_j^{\phi}(u)) d_u \right\}, \\ &s.t.\ \ \mathbb{E} \left\{ \sum_{j \in \mathcal{K}: X_j^{\phi}(t) \in \mathcal{N}_j^{\{0,1\}}} a_j^{\phi}(t) \right\} = 1. \end{aligned} \tag{7}$$

Seldom in the literature is it well explained that this means that the $a_j^{\phi}(t)$ become random variables, so that sometimes more than one server will be tagged simultaneously. This is not consistent with the framework of our original problem and is also unrealistic.

The linear constraint in (7) is covered by the introduction of a Lagrange multiplier ν.

$$\begin{aligned} \inf_{\nu} \sup_{\phi} \lim_{t \to +\infty} \frac{1}{t} \mathbb{E} \Bigg\{ \int_0^t \Bigg[&\sum_{j \in \mathcal{K}} R_j(X_j^{\phi}(u)) \\ &- \nu \sum_{j \in \mathcal{K}: X_j^{\phi}(t) \in \mathcal{N}_j^{\{0,1\}}} a_j^{\phi}(u) \Bigg] d_u \Bigg\} + \nu. \end{aligned} \tag{8}$$

For a given ν, we can now decompose (8) into K subproblems:

$$\sup_{\phi} \lim_{t\to+\infty} \frac{1}{t}\mathbb{E}\left\{\int_0^t \left[R_j(X_j^{\phi}(u)) - \nu a_j^{\phi}(u)\right] d_u\right\}, \quad (9)$$

where $a_j^{\phi}(u) = 0$ when $X_j^{\phi}(u) \in \mathcal{N}_j^{\{0\}}$, for $0 < u < t$, $j \in \mathcal{K}$.

In [32], Whittle defined a ν-subsidy policy for a project (server) as an optimal solution for (9), which provides the set of states where the given project will be passive (untagged), and introduced the following definition.

Definition 1. *Let $D(\nu)$ be the set of passive states of a project under a ν-subsidy policy. The project is indexable if $D(\nu)$ increases monotonically from $\emptyset$ to the set of all possible states for the project as ν increases from $-\infty$ to $+\infty$.*

In particular, if a project (server) j is indexable and there is a ν^* satisfying $n_j \notin D(\nu)$ for $\nu \le \nu^*$ and $n_j \in D(\nu)$ otherwise then this ν^* is the value of Whittle's index for project (server) j at state n_j. Whittle's index policy for the multi-queue system chooses a controllable server (a server in controllable states) with highest Whittle's index to be tagged (with others untagged) at each decision making epoch.

C. Indexability

We give the closed form of the optimal solution for Problem (9); that is, the Whittle's index policy, for the case with exponentially distributed job sizes. Our approach uses the theory of semi-Markov decision processes and the Hamilton-Jacobi-Bellman equation. Formulation in this way requires the exponential job size assumption.

Let $V_j^{\phi_j,\nu}(n_j, R_j)$ be, for policy ϕ_j, the expected value of the cumulative reward of a process for server $j \in \mathcal{K}$ that starts from state $n_j \in \mathcal{N}_j$ and ends when it first goes into an absorbing state $n_j^0 \in \mathcal{N}_j$ with reward rate $R_j(n_j) - \nu a_j^{\phi_j}(n_j)$. In particular, $V_j^{\phi_j,\nu}(n_j^0) = 0$ for any ϕ_j. Here, ϕ_j is a stationary policy for server j, which determines whether it is tagged or not according to its current state $X_j^{\phi_j}(t)$. Because state 0 is reachable from all other states, we can assume without loss of generality that $n_j^0 = 0$ for all $j \in \mathcal{K}$. For this section, we define $R_j(n_j) = \mu_j(n_j) - e^*\varepsilon_j(n_j)$, $j \in \mathcal{K}$, where e^* is defined as in (5).

Now, let $\mathcal{P}_j^H$, $j \in \mathcal{K}$, represent a process for server j that starts from state 0 until it reaches state 0 again, where ϕ_j is constrained to those policies satisfying $a_j^{\phi_j}(0) = 1$. The set of all such policies is denoted by Φ_j^H. It follows from [53, Corollary 6.20 and Theorem 7.5] that the average reward of process $\mathcal{P}_j^H$ is equivalent to the long-run average reward of the system.

Now an application of the *g-revised* criterion [53, Theorem 7.6, Theorem 7.7], yields the followed corollary to these two theorems.

Corollary 1. *For a server j as defined in Section III and a given $\nu < +\infty$, let $R_j(n_j) = \mu_j(n_j) - e^*\varepsilon_j(n_j) < +\infty$, there exists a real g, with $R_j^g(n_j) = R_j(n_j) - g$ such that if policy $\phi_j^* \in \Phi_j^H$ maximizes $V_j^{\phi_j,\nu}(n_j, R_j^g)$ then, ϕ_j^* also maximizes the long-run average reward of server j with reward rate $R_j(n_j) - a_j^{\phi_j^*}(n_j)\nu$, $n_j \in \mathcal{N}_j$, among all policies in Φ_j^H. In particular, this value of g, denoted by g^*, is equivalent to the maximized long-run average reward.*

In other words, if we compare the maximized average reward of process $\mathcal{P}_j^H$ under policy ϕ_j^* and policy ϕ_j^0 with $a_j^{\phi_j^0}(0) = 0$ (and all the actions for non-zero states are the same as ϕ_j^*), then the one with higher average reward is the optimal policy for (9). Note that, in our server farm model, if $a_j^{\phi_j^0}(0) = 0$, the actions for non-zero states are meaningless since the corresponding server (queue) will never leave state 0.

We start by finding ϕ_j^*. Let $V_j^{\nu}(n_j, R_j^g) = \sup_{\phi_j} V_j^{\phi_j,\nu}(n_j, R_j^g)$. The maximization of $V_j^{\phi_j,\nu}(n_j, R_j^g)$ can be written using the Hamilton-Jacobi-Bellman equation as

$$\begin{aligned} &V_j^{\nu}(n_j, R_j^g) \\ &= \max\left\{\left(R_j^g(n_j) - \nu\right)\tau_j^1(n_j) + \sum_{n\in\mathcal{N}_j} P_j^1(n_j, n)V_j^{\nu}(n, R_j^g),\right. \\ &\qquad \left. R_j^g(n_j)\tau_j^0(n_j) + \sum_{n\in\mathcal{N}_j} P_j^0(n_j, n)V_j^{\nu}(n, R_j^g)\right\}, \quad (10) \end{aligned}$$

where $\tau_j^1(n_j)$ and $\tau_j^0(n_j)$, are the expected sojourn time in state n_j for $a_j^{\phi_j}(n_j) = 1$, and $a_j^{\phi_j}(n_j) = 0$, respectively, and $P^1(n_j, n)$ and $P^0(n_j, n)$, $n_j, n \in \mathcal{N}_j$, are the transition probability for $a_j^{\phi_j}(n_j) = 1$ and $a_j^{\phi_j}(n_j) = 0$, respectively.

For (10), there is a specific ν, referred to as $\nu_j^*(n_j, R_j^g)$, satisfying

$$\begin{aligned} &\nu_j^*(n_j, R_j^g)\tau_j^1(n_j) \\ &= \sum_{n\in\mathcal{N}_j} P_j^1(n_j, n)V_j^{\nu}(n, R_j^g) - \sum_{n\in\mathcal{N}_j} P_j^0(n_j, n)V_j^{\nu}(n, R_j^g) \\ &\qquad + R_j^g(n_j)(\tau_j^1(n_j) - \tau_j^0(n_j)). \quad (11) \end{aligned}$$

For an indexable server j, we define a policy as follows:

$$\begin{aligned} &\text{if } \nu < \nu_j^*(n_j, R_j^g),\ j \text{ will be tagged} \\ &\text{if } \nu > \nu_j^*(n_j, R_j^g),\ j \text{ will be untagged, and} \\ &\text{if } \nu = \nu_j^*(n_j, R_j^g),\ j \text{ can be either tagged or untagged.} \end{aligned} \quad (12)$$

The $\nu^*(n_j, R_j^g)$, $n_j \in \mathcal{N}_j$, $j \in \mathcal{K}$, constitute *Whittle's index* [32] in this context, and (12) defines the optimal solution for Problem (9). According to (11), although the value of $\nu_j^*(n_j, R_j^g)$ may appear to rely on ν, we will prove later that, in our case, the value of $\nu_j^*(n_j, R_j^g)$ can be expressed in close form and is independent of ν, and that the server farm in our context is *indexable* according to the definition in [32].

Proposition 1. *For the system defined in Section III, $j \in \mathcal{K}$,*

$$\nu_j^*(n_j, R_j^g) = \frac{\lambda(\mu_j - e^*\varepsilon_j - g)}{\mu_j},\ n_j = 1, 2, \ldots, B_j - 1. \quad (13)$$

Proof. The proof is given in Appendix A. ■

The optimal policy, denoted by ϕ_j^*, that maximizes $V_j^{\phi_j,\nu}(n_j, R_j^g)$ also maximizes the average reward of process $\mathcal{P}_j^H$ with the value of g specified in Corollary 1 among all policies in Φ_j^H. For the optimal ν-subsidy policy, it remains to compare ϕ_j^* with $a_j^{\phi_j^*}(0) = 1$ and ϕ_j^0 with $a_j^{\phi_j^0}(0) = 0$.

Proposition 2. *For the system defined in Section III, $j \in \mathcal{K}$,*

$$\nu_j^*(0, R_j^g) = \frac{\lambda}{\mu_j}\left(\mu_j - e^*\varepsilon_j + e^*\varepsilon_j^0\right). \tag{14}$$

Proof. The proof is given in Appendix B. ■

Now Proposition 3 is a consequence of Propositions 1 and 2.

Proposition 3. *For the system defined in Section III, if job-sizes are exponentially distributed then the Whittle's index of server j at state n_j is:*

$$\nu_j^*(n_j, R_j^g) = \lambda(1 - e^*\frac{\varepsilon_j - \varepsilon_j^0}{\mu_j}),\ n_j = 0, 1, \ldots, B_j - 1.$$

Evidently then, the system is indexable.

Proof. The proof is given in Appendix C. ■

It is clear that the Whittle's index policy, which prioritizes server(s) with the highest index value at each decision making epoch, is equivalent to our MAIP policy defined in (4), when job sizes are exponentially distributed. Note that the form of Whittle's index in the case for general job-size distributions remains unclear. In Section VI, we demonstrate numerically the sensitivity of MAIP to highly varying job sizes.

D. Asymptotic Optimality

We will prove the asymptotic optimality of MAIP as the number of servers becomes large when job sizes are exponentially distributes and the number of servers is scaled under appropriate and reasonable conditions for large server farms (as discussed in Section I).

We will apply the proof methodology of Weber and Weiss [45] for the asymptotic optimality of index policies to our problem though, as we have already stated in Section II, this proof cannot be directly applied to our problem because of the presence of uncontrollable states. We define an additional (virtual) server, designated as server $K+1$, to handle the blocking case when all actual servers are full; this server has only one state (server $K+1$ never changes state) with zero reward rate. This virtual server is only used in the proof of asymptotic optimality in this section. For this server, $|\mathcal{N}_{K+1}^{\{0,1\}}| = 1$ and $\mathcal{N}^{\{0\}} = \emptyset$. In addition, we define $\mathcal{K}^+ = \mathcal{K} \cup \{K+1\}$ as the set of servers including this extra zero-reward server. The set of controllable states of these $K+1$ servers is defined as $\tilde{\mathcal{N}}^{\{0,1\}} = \bigcup_{j\in\mathcal{K}^+} \mathcal{N}_j^{\{0,1\}}$, and the set of uncontrollable states is $\tilde{\mathcal{N}}^{\{0\}} = \bigcup_{j\in\mathcal{K}^+} \mathcal{N}_j^{\{0\}}$.

In this section, those servers with identical buffer size, service rate, and energy consumption rate are grouped as a server group, and we label these server groups as server groups $1, 2, \ldots, \tilde{K}$. For servers i, j of the same server group, $\mathcal{N}_i^{\{0,1\}} = \mathcal{N}_j^{\{0,1\}}$ and $\mathcal{N}_i^{\{0\}} = \mathcal{N}_j^{\{0\}}$. For clarity of presentation, we define $\tilde{\mathcal{N}}_i^{\{0,1\}}$ and $\tilde{\mathcal{N}}_i^{\{0\}}$, $i = 1, 2, \ldots, \tilde{K}$ as, respectively, the sets of controllable and uncontrollable states of servers in server group i. We regard states for different server groups as different states; that is, $\tilde{\mathcal{N}}_j^{\{0,1\}} \cap \tilde{\mathcal{N}}_i^{\{0,1\}} = \emptyset$ and $\tilde{\mathcal{N}}_j^{\{0\}} \cap \tilde{\mathcal{N}}_i^{\{0\}} = \emptyset$ for different server groups i and j, $i, j = 1, 2, \ldots, \tilde{K}$. Let $Z_i^\phi(t)$ be the random variable representing the proportion of servers in state $i \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}$ at time t under policy ϕ. As previously, we label states $i \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}$ as $1, 2, \ldots, I$, where $I = |\tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}|$, and use $\mathbf{Z}^\phi(t)$ to denote the random vector $(Z_1^\phi(t), Z_2^\phi(t), \ldots, Z_I^\phi(t))$. Correspondingly, actions $a_j^\phi(n_j)$, $n_j \in \mathcal{N}_j$, $j \in \mathcal{K}^+$, correspond to actions $a^\phi(i)$, $i \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}$.

Let $\mathbf{z}, \mathbf{z}' \in R^I$ be instantiations of $\mathbf{Z}^\phi(t)$, $t > 0$, $\phi \in \Phi$. Transitions of the random vector $\mathbf{Z}^\phi(t)$ from $\mathbf{z}$ to $\mathbf{z}'$ can be written as $\mathbf{z}' = \mathbf{z} + \mathbf{e}_{i,i'}$, where $\mathbf{e}_{i,i'}$ is a vector of which the ith element is $+\frac{1}{K+1}$, the i'th element is $-\frac{1}{K+1}$ and otherwise is zero, $i, i' \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}$. In particular, for the server farm defined in Section III, servers in server group j only appear in state $i \in \tilde{\mathcal{N}}_j^{\{0,1\}} \cup \tilde{\mathcal{N}}_j^{\{0\}}$; that is, the transition from $\mathbf{z}$ to $\mathbf{z}' = \mathbf{z} + \mathbf{e}_{i,i'}$, $i \in \tilde{\mathcal{N}}_j^{\{0,1\}} \cup \tilde{\mathcal{N}}_j^{\{0\}}$, $i' \in \tilde{\mathcal{N}}_{j'}^{\{0,1\}} \cup \tilde{\mathcal{N}}_{j'}^{\{0\}}$, $j, j' = 1, 2, \ldots, \tilde{K}$, $j \neq j'$ never occur. We address such impossible transitions by setting the corresponding transition probabilities to zero. The states $i \in \tilde{\mathcal{N}}^{\{0,1\}}$ are ordered according to descending index values, where all states $i \in \tilde{\mathcal{N}}^{\{0\}}$ follow the controllable states in the ordering, with $a^\phi(i) = 0$ for $i \in \tilde{\mathcal{N}}^{\{0\}}$. Then, we set the state $i \in \mathcal{N}_{K+1}^{\{0,1\}}$ of the zero-reward server, which is also a controllable state, to come after all the other controllable states but to precede the uncontrollable states. Because of the existence of the zero-reward server $K+1$, the number of servers in controllable states can always meet the constraint (7). Note here that we artificially move the state of server $K+1$ and the uncontrollable states to places in the ordering that do not accord with their indices, which are zero. We will show later that such movements do not affect the long-run average performance of Whittle's index policy, which exists and is equivalent to MAIP in our context. The position of a state in the ordering $i = 1, 2, \ldots, I$ is also defined as its label.

Let $\gamma^{OR}(\phi)$ be the long-run average reward of the *original problem* (6) (that is, without relaxation) under policy ϕ, and $\gamma^{LR}(\phi)$ the long-run average reward of the *relaxed problem* (7) under policy ϕ. In addition, let $\gamma^{OR} = \max_\phi\{\gamma^{OR}(\phi)\}$, the maximal long-run average reward of the original problem, and $\gamma^{LR} = \max_\phi\{\gamma^{LR}(\phi)\}$, the maximal long-run average reward of the relaxed problem. From the definition of our system, $\gamma^{LR}(\phi)/K, \gamma^{OR}(\phi)/K \leq \max_{j\in\mathcal{K}, n_j\in\mathcal{N}_j} R_j(n_j) < +\infty$, where $R_j(n_j)$ is the reward rate of server j in state n_j as defined before. Let $index$ denote the Whittle's index policy, then, we obtain $\gamma^{OR}(index)/K \leq \gamma^{OR}/K \leq \gamma^{LR}/K$. Following the idea of [45], we prove, under Whittle's index policy, that $\gamma^{OR}(index)/K - \gamma^{LR}/K \to 0$ when K is scaled in a certain way.

To demonstrate asymptotic optimality, we now describe the stationary policies, including Whittle's index policy, in another way. Let $u_i^\phi(\mathbf{z}) \in [0, 1]$, $\mathbf{z} \in R^I$, $i = 1, 2, \ldots, I$, be the probability for a server in state $i \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}$ to be tagged ($a^\phi(i) = 1$) when $\mathbf{Z}^\phi(t) = \mathbf{z}$. Then, $1 - u_i^\phi(\mathbf{z})$ is the probability

for a server in state i to be untagged ($a^\phi(i) = 0$).

Define $\mathcal{N}_i^+$, $i \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}$ to be the set of states that precede state i in the ordering. Then, for Whittle's index policy, we obtain

$$u_i^{index}(\mathbf{z}) = \frac{1}{z_i} \min \left\{ z_i, \max \left\{ 0, \frac{1}{K+1} - \sum_{i' \in \mathcal{N}_i^+} z_{i'} \right\} \right\}. \tag{15}$$

Our multi-queue system is stable, since any stationary policy will lead to an irreducible Markov chain for the associated process and the number of states is finite. Then, for a policy $\phi \in \Phi$, the vector $\mathbf{X}^\phi(t)$ converges as $t \to \infty$ in distribution to a random vector $\mathbf{X}^\phi$. In the equilibrium region, let π_j^ϕ be the steady state distribution of X_j^ϕ for server j, $j \in \mathcal{K}^+$, under policy $\phi \in \Phi$, where $\pi_j^\phi(i)$, $i \in \mathcal{N}_j$, is the steady state probability of state i. For clarity of presentation, we extend vector π_j^ϕ to a vector of length I, written $\tilde{\pi}_j^\phi$, of which the ith element is $\pi_j^\phi(i)$, if $i \in \mathcal{N}_j$, and otherwise, 0. The long-run expected value of $\mathbf{Z}^\phi(t)$ is $\sum_{j=1}^{K+1} \tilde{\pi}_j^\phi/(K+1)$. In the server farm defined in Section III, the long-run expected value of $\mathbf{Z}^\phi(t)$ will be a member of the set

$$\mathcal{Z} = \left\{ \mathbf{z} \in R^I | \sum_{i \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}} z_i \equiv 1, \right. \\ \left. \forall i \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}},\ z_i \geq 0 \right\}. \tag{16}$$

Write $q^1(\mathbf{z}, z_i, z_i')$ and $q^0(\mathbf{z}, z_i, z_i')$, $\mathbf{z} \in \mathcal{Z}$, $i \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}$, as the average transition rate of the ith element in vector $\mathbf{z}$ from z_i to z_i', under tagged and untagged actions, respectively. Then, the average transition rate of the ith element of $\mathbf{z}$ under policy ϕ is given by

$$q^\phi(\mathbf{z}, z_i, z_i') = u_i^\phi(\mathbf{z}) q^1(\mathbf{z}, z_i, z_i') + (1 - u_i^\phi(\mathbf{z})) q^0(\mathbf{z}, z_i, z_i').$$

We consider the following differential equation for a stochastic process, denoted by $\mathbf{z}^\phi(t) = (z_1^\phi(t), z_2^\phi(t), \ldots, z_I^\phi(t))$,

$$\frac{dz_i^\phi(t)}{dt} = \sum_{z_i'} \Big[z_i'(t) q^\phi(\mathbf{z}^\phi(t), z_i', z_i^\phi(t)) \\ - z_i^\phi(t) q^\phi(\mathbf{z}^\phi(t), z_i^\phi(t), z_i') \Big]. \tag{17}$$

Because of the global balance at an equilibrium point of $\lim_{t\to+\infty} \int_0^t \mathbf{z}^\phi(u)du/t$, if exists, denoted by $\mathbf{z}^\phi$, $d\mathbf{z}^\phi(t)/dt|_{\mathbf{z}^\phi(t)=\mathbf{z}^\phi} = \mathbf{0}$. Let OPT represent the optimal solution of the relaxed problem (7) and recall that $index$ represents the Whittle's index policy. Since $u_i^{index}(\mathbf{z}^{index}) = u_i^{OPT}(\mathbf{z}^{index})$, following the proof of [45, Theorem 2], we obtain $d\mathbf{z}^{OPT}(t)/dt|_{\mathbf{z}^{OPT}(t)=\mathbf{z}^{index}} = \mathbf{0}$ and $\mathbf{z}^{index} = \mathbf{z}^{OPT}$, if both $\mathbf{z}^{index}$ and $\mathbf{z}^{OPT}$ exist. The existence of $\mathbf{z}^{index}$ and $\mathbf{z}^{OPT}$ will be discussed later.

For a small $\delta > 0$, we define $\overline{R}^{\delta,\phi}$ as the average reward rates during the time period that $|\mathbf{Z}^\phi(t) - \mathbf{z}^\phi(t)| \leq \delta$ under policy ϕ with $\mathbf{Z}^\phi(0) = \mathbf{z}^\phi(0)$, and $R_m/K = \sup_\phi \limsup_{t\to+\infty} |R(\mathbf{X}^\phi(t))/K| < +\infty$ is an upper bound of the absolute value of the reward rate divided by K. Then,

$$\frac{\gamma^{LR}}{K} - \frac{\gamma^{OR}(index)}{K} \\ \leq \lim_{\delta\to 0} \lim_{t\to+\infty} \frac{1}{t} \int_0^t \left[\frac{R_m}{K} P\left\{ |\mathbf{Z}^{OPT}(u) - \mathbf{z}^{OPT}(t)| > \delta \right\} \right. \\ + \frac{R_m}{K} P\left\{ |\mathbf{Z}^{index}(u) - \mathbf{z}^{index}(t)| > \delta \right\} \\ + \frac{\overline{R}^{\delta,OPT}}{K} P\left\{ |\mathbf{Z}^{OPT}(u) - \mathbf{z}^{OPT}(t)| \leq \delta \right\} \\ \left. - \frac{\overline{R}^{\delta,index}}{K} P\left\{ |\mathbf{Z}^{index}(u) - \mathbf{z}^{index}(t)| \leq \delta \right\} \right] du. \tag{18}$$

The server farm is decomposed into $\tilde{K}$ server groups, with number of servers in the ith group denoted by K_i, $i = 1, 2, \ldots, \tilde{K}$. Then, $K = \sum_{i=1}^{\tilde{K}} K_i$. Following the proof of [45, Proposition], for any $K_i = K_i^0 n$, $K_i^0 = 1, 2, \ldots$, $i = 1, 2, \ldots, \tilde{K}$, $n = 1, 2, \ldots$, $\delta > 0$ and ϕ is set to be either $index$ or OPT,

$$\lim_{n\to+\infty} \lim_{t\to+\infty} \frac{1}{t} \int_0^t P\{|\mathbf{Z}^\phi(u) - \mathbf{z}^\phi(u)| > \delta\} du = 0. \tag{19}$$

We provide a justification of (19) in Appendix D, following [54, Chapter 7]. Then, as $n \to +\infty$, the existence of an equilibrium point of $\lim_{t\to+\infty} \int_0^t \mathbf{Z}^\phi(u)du/t$ leads to the existence of $\mathbf{z}^\phi = \lim_{t\to+\infty} \int_0^t \mathbf{z}^\phi(u)du/t$ (using the Lipschitz continuity of the right side of Equation (17) as a function of $\mathbf{z}^\phi(t)$). We obtain

$$\lim_{n\to+\infty} \lim_{\delta\to 0} \left(\frac{\overline{R}^{\delta,OPT}}{K} - \frac{\overline{R}^{\delta,index}}{K} \right) = 0,$$

and

$$\lim_{n\to+\infty} \left(\frac{\gamma^{LR}}{K} - \frac{\gamma^{OR}(index)}{K} \right) = 0. \tag{20}$$

Finally, $\gamma^{OR}(index)/K - \gamma^{OR}/K \to 0$ as $n \to +\infty$; that is, MAIP (Whittle's index policy) approaches the optimal solution in terms of energy efficiency as the number of servers in each server group tends to infinity at the appropriate rate.

VI. Numerical Results

In this section, we provide extensive numerical results obtained by simulation to evaluate the performance of the MAIP policy. All results are presented in the form of an observed mean from multiple independent runs of the corresponding experiment. The confidence intervals at the 95% level based on the Student's t-distribution are maintained within $\pm 5\%$ of the observed mean. For convenience of describing the results, given two numerical quantities $x > 0$ and $y > 0$, we define the *relative difference* of x to y as $(x - y)/y$.

In all experiments, we have a system of servers that are divided into three server groups. Servers in each server group i, $i = 1, 2, 3$, have the same buffer size, service rate and energy consumption rate, denoted by $\bar{B}_i$, $\bar{\mu}_i$, $\bar{\varepsilon}_i$ and $\bar{\varepsilon}_i^0$, respectively. We consider this to be a realistic setting since in practice a server farm is likely to comprise multiple servers of the same

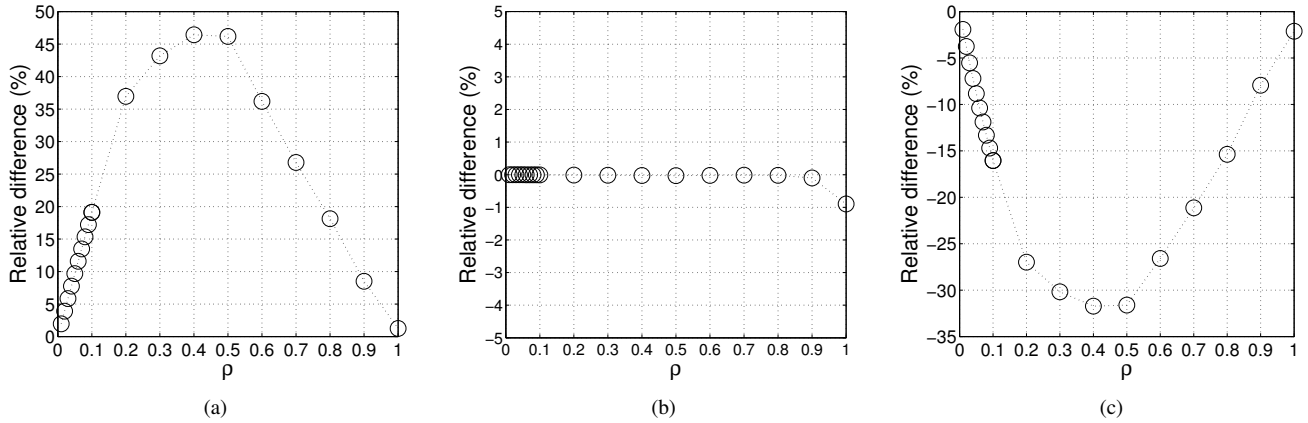


Fig. 1. Performance comparison with respect to the normalized offered traffic ρ. (a) Relative difference of $\mathcal{L}^{\mathrm{MAIP}}/\mathcal{E}^{\mathrm{MAIP}}$ to $\mathcal{L}^{\mathrm{MNIP}}/\mathcal{E}^{\mathrm{MNIP}}$. (b) Relative difference of $\mathcal{L}^{\mathrm{MAIP}}$ to $\mathcal{L}^{\mathrm{MNIP}}$. (c) Relative difference of $\mathcal{E}^{\mathrm{MAIP}}$ to $\mathcal{E}^{\mathrm{MNIP}}$.

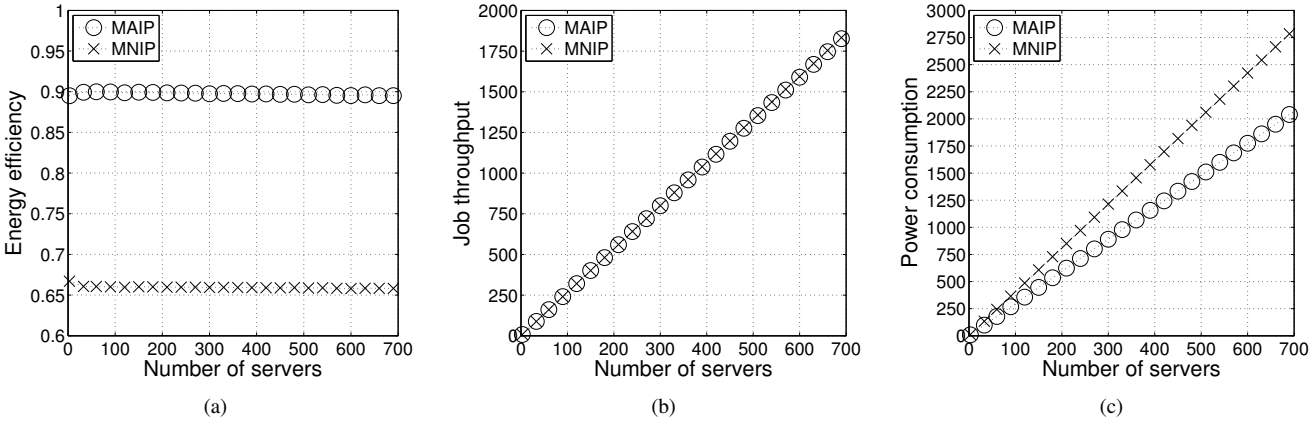


Fig. 2. Performance comparison with respect to the number of servers K. (a) Energy efficiency. (b) Job throughput. (c) Power consumption.

type purchased at a time. If not otherwise specified, we assume that job sizes are exponentially distributed.

Recall that, as defined in Section III, the job throughput is the average job departure rate (jobs per second), the power consumption is the average energy consumption rate (Watt), and the energy efficiency is the ratio of job throughput to power consumption (jobs per Watt second). Also, we have normalized the average job size to one (Byte) in Section III.

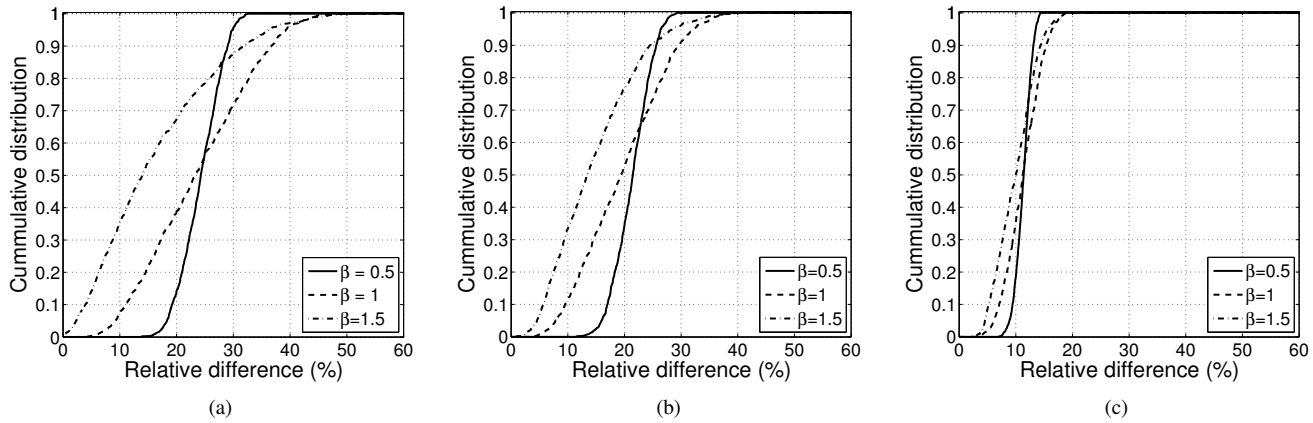
A. Effect of Idle Power

Recall that MAIP is designed to take into account the effect of idle power. To demonstrate the effect of idle power on job assignment, here we consider a baseline policy, called *Most energy-efficient available server first Neglecting Idle Power* (MNIP). As its name suggests, MNIP is a straightforward variant of MAIP that neglects idle power and hence treats $\varepsilon_j^0 = 0$ for all $j \in \mathcal{K}$ in the process of selecting servers for job assignment. We compare MAIP with MNIP in terms of energy efficiency, job throughput and power consumption under various system parameters.

For the set of experiments in Fig. 1, each server group has 15 servers, where we set $\bar{B}_i = 10$ and $\bar{\varepsilon}_i^0/\bar{\varepsilon}_i = 0.4i - 0.3$ for $i = 1, 2, 3$, and randomly generate $\bar{\mu}_i$ and $\bar{\varepsilon}_i$ as $\bar{\mu}_1 = 6.86$, $\bar{\varepsilon}_1 = 6.86$, $\bar{\mu}_2 = 3.64$, $\bar{\varepsilon}_2 = 3.72$, $\bar{\mu}_3 = 2.87$, $\bar{\varepsilon}_3 = 3.15$. The normalized offered traffic ρ is varied from 0.01 to 0.9.

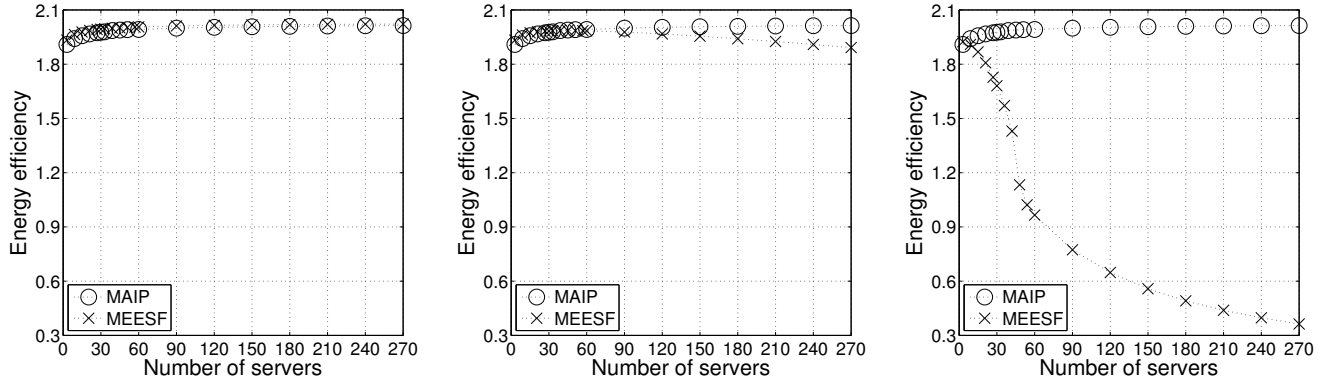
The results in Fig. 1 are presented in the form of the relative difference of MAIP to MNIP in terms of each corresponding performance measure. We observe in Fig. 1(b) that, in all cases, both policies have almost the same performance in job throughput. We also observe in Fig. 1(a) and Fig. 1(c) that, in the case where $\rho \to 0$ or $\rho \to 1$, the two policies are close to each other in terms of both energy efficiency and power consumption. This is because in such trivial and extreme cases the system is at all times either almost empty or almost fully occupied. However, in the realistic cases where ρ is not too large and not too small, MAIP significantly outperforms MNIP with a gain of over 45% in energy efficiency at $\rho = 0.4, 0.5$.

In Fig. 2, we use the same settings as in Fig. 1, except that we fix the normalized offered traffic ρ at 0.6 and vary the number of servers K from 3 to 690. Note that here we increase K by increasing the number of servers in each of the three server groups. We observe in Fig. 2(b) that, under such a medium traffic load, the service capacity is sufficiently large, so that with both MAIP and MNIP almost all jobs can be admitted and hence the job throughput is almost identical to the arrival rate for all values of K. As a result,



(a) (b) (c)

Fig. 3. Cumulative distribution of the relative difference of $\mathcal{L}^{\mathrm{MAIP}}/\mathcal{E}^{\mathrm{MAIP}}$ to $\mathcal{L}^{\mathrm{MNIP}}/\mathcal{E}^{\mathrm{MNIP}}$. (a) $\rho = 0.4$. (b) $\rho = 0.6$. (c) $\rho = 0.8$.



(a) (b) (c)

Fig. 4. Performance comparison in terms of the energy efficiency with respect to the number of servers K. (a) $\Delta = 0$. (b) $\Delta = 0.0005$. (c) $\Delta = 0.01$.

the job throughput increases almost linearly with respect to the number of servers K. We observe in Fig. 2(c) that, for both policies, the power consumption also increases almost linearly with respect to the number of servers K. However, it is clear that the power consumption of MAIP increases at a significantly smaller rate than that of MNIP, which results in a substantial improvement of the energy efficiency by nearly 36% in all cases as seen in Fig. 2(a).

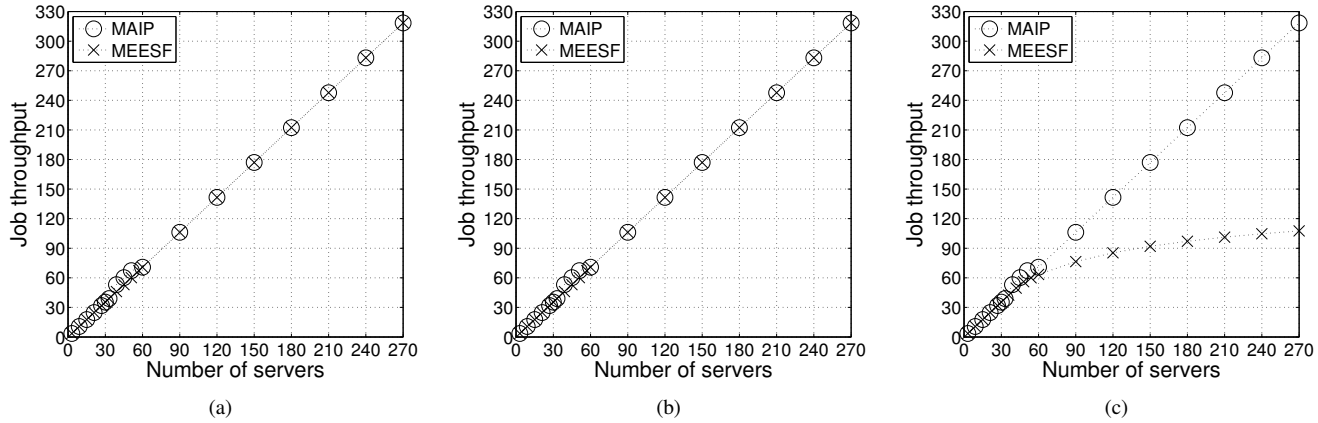
For the set of experiments in Fig. 3, we again consider a system where each server group has 15 servers and we set $\bar{B}_i = 10$ for $i = 1, 2, 3$. We introduce a parameter β, where different values of β lead to different levels of server heterogeneity. In particular, we consider three different values for β, i.e., $\beta = 0.5, 1, 1.5$. We set $\bar{\varepsilon}_i^0/\bar{\varepsilon}_i = (0.4i - 0.3)^\beta$ for $i = 1, 2, 3$. The set of service rates $\bar{\mu}_i$ are randomly generated from the range $[1, 10]$ and are arranged in a non-increasing order, i.e., $\bar{\mu}_1 \geq \bar{\mu}_2 \geq \bar{\mu}_3$. For the set of energy consumption rates $\bar{\varepsilon}_i$, we first choose two real numbers $\bar{a}_1$ and $\bar{a}_2$ randomly from $[0.5, 1]$. Then, with $\bar{\mu}_1/\bar{\varepsilon}_1 = 200$, we set $\bar{\mu}_i/\bar{\varepsilon}_i = \bar{a}_{i-1}^\beta \bar{\mu}_{i-1}/\bar{\varepsilon}_{i-1}$ for $i = 2, 3$.

The results in Fig. 3 are obtained from 1000 experiments and are plotted in the form of cumulative distribution of the relative difference of MAIP to MNIP in terms of the energy efficiency. We observe in Fig. 3 that MAIP significantly outperforms MNIP by up to 60%. It can also be observed from Fig. 3(a) and Fig. 3(b) that MAIP outperforms MNIP by more than 10% in nearly 100% of the experiments for the case $\beta = 0.5$. In addition, we observe in Fig. 3 that, as the level of server heterogeneity (i.e., the value of β) becomes higher, the performance improvement of MAIP over MNIP in general becomes larger, although the gain is decreasing when the normalized offered traffic ρ approaches 0.8, similar to what is observed in Fig. 1(a).

B. Effect of Jockeying Cost

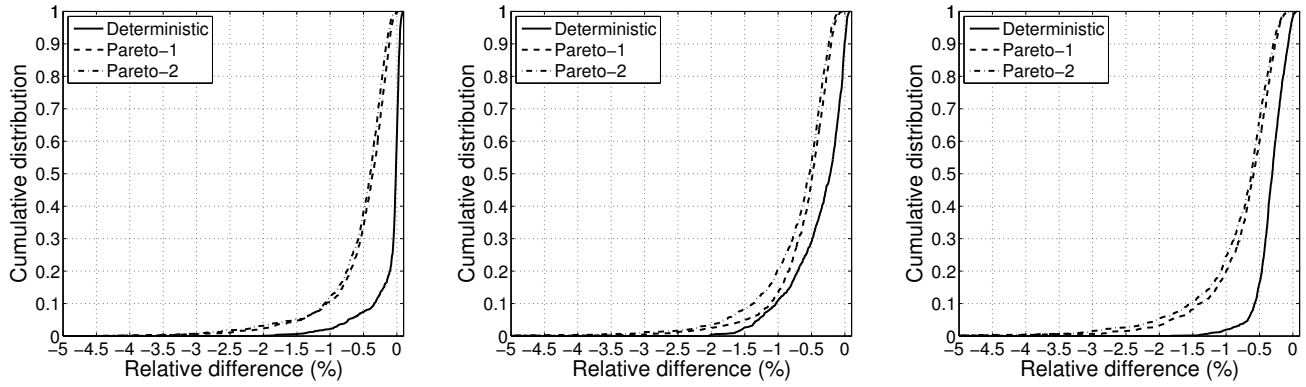
Recall that MAIP is designed as a non-jockeying policy, which is more appropriate than jockeying policies for job assignment in a large-scale server farm. As discussed in Section II, jockeying policies suit a small server farm where the cost associated with jockeying is negligible. In large-scale systems, the cost associated with jockeying can be significant and may have a snowball effect on the system performance. Here, we demonstrate the benefits of MAIP in a server farm where jockeying costs are high, by comparing it with a jockeying policy known as *Most Energy Efficient Server First* (MEESF) proposed in [20].

The settings of servers in each of the three server groups are based on the benchmark results of Dell PowerEdge rack



(a) (b) (c)

Fig. 5. Performance comparison in terms of the job throughput with respect to the number of servers K. (a) $\Delta = 0$. (b) $\Delta = 0.0005$. (c) $\Delta = 0.01$.



(a) (b) (c)

Fig. 6. Sensitivity of the energy efficiency of MAIP to the job-size distribution. (a) $\rho = 0.4$. (b) $\rho = 0.6$. (c) $\rho = 0.8$.

servers R610 (August 2010), R620 (May 2012) and R630 (April 2015) [55]. Specifically, we normalize $\bar{\mu}_3$ and $\bar{\varepsilon}_3$ to one and then set $\bar{\mu}_1/\bar{\mu}_3 = 3.5$, $\bar{\varepsilon}_1/\bar{\varepsilon}_3 = 1.2$, $\bar{\varepsilon}_1^0/\bar{\varepsilon}_1 = 0.2$, $\bar{\mu}_2/\bar{\mu}_3 = 1.4$, $\bar{\varepsilon}_2/\bar{\varepsilon}_3 = 1.1$, $\bar{\varepsilon}_2^0/\bar{\varepsilon}_2 = 0.2$ and $\bar{\varepsilon}_3^0/\bar{\varepsilon}_3 = 0.3$. We also set $\bar{B}_i = 10$ for $i = 1, 2, 3$ and $\rho = 0.6$. The number of servers K is varied from 3 to 270, where we increase K by increasing the number of servers in each of the three server groups.

Let us assume that each jockeying action incurs a (constant) delay Δ. That is, when a job is reassigned from server i to server j, it will be suspended for a period Δ before resumed on server j. Clearly, when $\Delta > 0$, this is equivalent to increasing the size of the job and hence its service requirement. Accordingly, for a given system, a non-zero cost per jockeying action indeed increases the traffic load. We consider three different values for Δ. The case where $\Delta = 0$ is for zero jockeying cost, the case where $\Delta = 0.0005$ indicates a relatively small cost per jockeying action, and the case where $\Delta = 0.01$ represents a large cost per jockeying action. The results are presented in Fig. 4 for the energy efficiency and in Fig. 5 for the job throughput.

For the case where $\Delta = 0$, we have a similar observation in Fig. 5(a) to that in Fig. 2(b). That is, under a medium traffic load, the service capacity is sufficiently large, so that both MAIP and MEESF yield a job throughput that is almost identical to the arrival rate for all values of K. We observe in Fig. 4(a) that, in this case, MEESF consistently outperforms MAIP in terms of the energy efficiency, even though with a very small margin.

For the case where $\Delta = 0.0005$, we observe in Fig. 5(b) that, since the cost per jockeying action is relatively small, the service capacity turns out to be sufficiently large so that the job throughput of MEESF is not affected. We also observe in Fig. 4(b) that, when the number of servers K is small, the energy efficiency of MEESF is still better than that of MAIP. However, when the number of servers K is large, the energy efficiency of MEESF is clearly degraded. This is because, as K increases, the number of jockeying actions required for a job on average increases. With a non-zero cost per jockeying action, it can substantially increase the power consumption, since we are forced to use more of those less energy-efficient servers to meet the increased traffic load.

The effect is more profound when Δ is increased to 0.01. In this case, as shown in Fig. 4(c) and Fig. 5(c), the cost associated with jockeying is so high that both the job throughput and the energy efficiency of MEESF are significantly degraded, due to the substantially increased traffic load.

C. Sensitivity to Job-Size Distribution

The workload characterizations of many computer science applications, such as Web file sizes, IP flow durations, and the lifetimes of supercomputing jobs, are known to exhibit heavy-tailed distributions [27], [28]. Here, we are interested to see if the performance of MAIP is sensitive to the job-size distribution. To this end, in addition to the exponential distribution, we further consider three different distributions, i.e., deterministic, Pareto with the shape parameter set to 2.001 (Pareto-1 for short), and Pareto with the shape parameter set to 1.98 (Pareto-2 for short). In all cases, we set the mean to be one.

We use the same experiment settings as in Fig. 3 with $\beta = 1$. In each experiment, we obtain the energy efficiency of MAIP, and compute the relative difference of the one using each corresponding distribution to the one using the exponential distribution. Fig. 6 plots the cumulative distribution of the relative difference results obtained from the 1000 experiments for each particular value of the normalized offered traffic ρ. We observe in Fig. 6 that all relative difference results are between -5% and 0. Given that the confidence intervals of these simulation results are maintained within $\pm 5\%$ of the observed mean with a 95% confidence level, the energy efficiency of MAIP seems not to be too sensitive to the job-size distribution.

VII. Conclusions

We have studied the stochastic job assignment problem in a server farm comprising multiple processor sharing servers with different service rates, energy consumption rates and buffer sizes. Our aim has been to maximize the energy efficiency of the entire system, defined as the ratio of the long-run average job departure rate to the long-run average power consumption, by effectively assigning jobs/requests to these servers. To this end, we have introduced MAIP job assignment policy and have proved its equivalence to the Whittle's index policy when job sizes are exponentially distributed. MAIP only requires information of binary states of servers, and can be implemented by using a binary variable for each server. This policy does not require any estimation or prediction of average arrival rate. MAIP has been proven to approach optimality as the numbers of servers in server groups tend to infinity when job sizes are exponentially distributed. This asymptotic property is appropriate to a large-scale server farm that is likely to purchase and upgrade a large number of servers with the same style and attributes at the same time. The proof for asymptotic optimality has been completed by applying the ideas of Weber and Weiss [45] to our multi-queue system with inevitable uncontrollable states. Extensive numerical results have illustrated the significant superiority of MAIP over MNIP (the baseline policy) in a general situation where energy efficiency of each server differs from its effective energy efficiency. MAIP has been shown numerically to give similar energy efficiency results in cases of exponential and Pareto job-size distributions, which indicates that it is appropriate to a server farm with highly varying job sizes. Through a numerical example, we have shown that MAIP is more appropriate than MEESF for a server farm with non-zero jockeying cost in the example, which is also useful for a real large-scale system with significant cost of job reassignment.

Appendix A
Proof of Proposition 1

Lemma 1. *For the system defined in Section III with exponentially distributed job sizes, let $a_j^{\nu,g}(n_j)$ denote the action taken in state n_j under the policy which optimizes (10) for a given ν. The following two statements are equivalent: (1) $\nu \le \lambda\left(V_j^\nu(n_j+1,R_j^g) - V_j^\nu(n_j,R_j^g)\right)$, and (2) $a_j^{\nu,g}(n_j) = 1$, where $n_j = 1,2,\ldots,B_j-1$.*

Proof. According to (10) and (11), observe that, for a given R_j^g, $n_j = 1,2,\ldots,B_j-1$, $j \in \mathcal{K}^+$, there exists a value $\nu_j^*(n_j,R_j^g) \in R$, satisfying

$$a_j^{\nu,g}(n_j) = \begin{cases} 1, & \nu \le \nu_j^*(n_j,R_j^g), \\ 0, & \text{otherwise.} \end{cases}$$

We rewrite (10) as

$$\begin{aligned} V_j^\nu(n_j,R_j^g) = \max\Bigg\{ & \frac{\mu_j - e^*\varepsilon_j - g}{\mu_j} + V^\nu(n_j-1,R_j^*), \\ & \frac{\mu_j - e^*\varepsilon_j - g}{\lambda+\mu_j} - \frac{\nu}{\lambda+\mu_j} + \frac{\lambda}{\lambda+\mu_j}V^\nu(n_j+1, \\ & R_j^*) + \frac{\mu_j}{\lambda+\mu_j}V^\nu(n_j-1,R_j^*)\Bigg\}, \end{aligned} \quad (21)$$

where $1/(\lambda+\mu_j)$ and $1/\mu_j$ are the average lengths of one sojourn in state n_j with and without new arrivals (tagged and untagged), respectively. We obtain

$$\begin{aligned} \nu_j^*(n_j,R_j^g) = {} & \lambda\left(V_j^\nu(n_j+1,R_j^g) - V_j^\nu(n_j-1,R_j^g)\right) \\ & - \frac{\lambda(\mu_j - e^*\varepsilon_j - g)}{\mu_j} \end{aligned} \quad (22)$$

Again, by (21), if $a_j^{\nu,g}(n_j) = 1$, then

$$\begin{aligned} & V_j^\nu(n_j,R_j^g) - V_j^\nu(n_j-1,R_j^g) \\ & = \frac{\lambda}{\mu_j}\left(V_j^\nu(n_j+1,R_j^g) - V_j^\nu(n_j,R_j^g)\right) + \frac{\mu_j - e^*\varepsilon_j - g}{\mu_j} - \frac{\nu}{\mu_j}. \end{aligned} \quad (23)$$

Recall that $a_j^{\nu,g}(n_j) = 1$ when $\nu \le \nu_j^*(n_j,R_j^g)$, where the identity indicates no difference between $a_j^{\nu,g}(n_j) = 0$ and $a_j^{\nu,g}(n_j) = 1$. We conclude that $\nu_j^*(n_j,R_j^g) = \lambda\left(V_j^\nu(n_j+1,R_j^g) - V_j^\nu(n_j,R_j^g)\right)$, $n_j = 1,2,\ldots,B_j-1$. This proves the lemma. ∎

Lemma 2. *For the system defined in Section III with exponentially distributed job sizes, $\forall n_j = 1,2,\ldots,B_j-1$, $j = 1,2,\ldots,K+1$, $V_j^\nu(n_j+1,R_j^g) - V_j^\nu(n_j,R_j^g) \ge \frac{\mu_j - e^*\varepsilon_j - g}{\mu_j}$.*

Proof. For $n_j = B_j - 1$, $V_j^\nu(B_j,R_j^g) - V_j^\nu(B_j-1,R_j^g) = (\mu_j - e^*\varepsilon_j - g)/\mu_j$.

According to (21), if $a_j^{\nu,g}(n_j) = 0$, $n_j = 1,2,\ldots,B_j-1$, then, $V_j^\nu(n_j,R_j^g) - V_j^\nu(n_j-1,R_j^g) = (\mu_j - e^*\varepsilon_j - g)/\mu_j$. Also, if $a_j^{\nu,g}(n_j) = 1$, $n_j = 1,2,\ldots,B_j-1$, then we obtain (23) and, based on Lemma 1,

$$V_j^\nu(n_j,R_j^g) - V_j^\nu(n_j-1,R_j^g) \ge \frac{\mu_j - e^*\varepsilon_j - g}{\mu_j}. \quad (24)$$

This proves the lemma. ■

Next, we complete the proof of Proposition 1.

Proof. If $a_j^{\nu,g}(n_j) = 0$ then, by Lemmas 1 and 2, $\nu > \lambda\left(V^\nu(n_j+1,R_j^g) - V^\nu(n_j,R_j^g)\right) \geq (\lambda(\mu_j - e^*\varepsilon_j - g))/\mu_j$. It remains to prove that, if $\nu > (\lambda(\mu_j - e^*\varepsilon_j - g))/\mu_j$, then $a_j^{\nu,g}(n_j) = 0$. By definition, $V_j^\nu(B_j,R_j^g) - V_j^\nu(B_j-1,R_j^g) = (\mu_j - e^*\varepsilon_j - g)/\mu_j$, and by Lemma 1, $\nu_j^*(B_j-1,R_j^g) = \lambda(\mu_j - e^*\varepsilon_j - g)/\mu_j$.

Finally, we complete the proof by induction. Assume that $\nu_j^*(n,R_j^g) = \lambda(\mu_j - e^*\varepsilon_j - g)/\mu_j$, for all $n \geq n_j$, $n_j = 2,3,\ldots,B_j-1$. If $\nu > \lambda(\mu_j - e^*\varepsilon_j - g)/\mu_j$, then $a_j^{\nu,g}(n) = 0$ for all $n \geq n_j$; that is, $V_j^\nu(n,R_j^g) - V_j^\nu(n-1,R_j^g) = (\mu_j - e^*\varepsilon_j - g)/\mu_j$, for all $n \geq n_j$. Together with Lemma 1, $\nu_j^*(n_j - 1,R_j^g) = \lambda\left(V_j^\nu(n_j,R_j^g) - V_j^\nu(n_j-1,R_j^g)\right) = \lambda(\mu_j - e^*\varepsilon_j - g)/\mu_j$. That is, if $\nu > \lambda(\mu_j - e^*\varepsilon_j - g)/\mu_j$, then $\nu > \nu_j^*(n_j - 1,R_j^g)$; that is, $a_j^{\nu,g}(n_j-1) = 0$. This proves the lemma. ■

APPENDIX B
PROOF OF PROPOSITION 2

Proof. We now dicuss the action made in state 0; that is, $a_j^{\phi_j}(0)$. Let π_{j,n_j} be the steady state distribution of state $n_j \in \mathcal{N}_j$ under policy ϕ_j^* over the process $\mathcal{P}_j^H$. We consider the following problem.

$$\max\left\{-e^*\varepsilon_j^0,\ (-e^*\varepsilon_j^0)\pi_{j,0} + (1-\pi_{j,0})(\mu_j - e^*\varepsilon_j) - \sum_{n_j=1}^{B_j-1} a_j^\nu(n_j)\pi_{j,n_j}\nu - \pi_{j,0}\nu\right\}. \quad (25)$$

It follows that $a_j^\nu(0) = 1$ is equivalent to

$$\nu \leq \frac{(1-\pi_{j,0})(\mu_j - e^*\varepsilon_j + e^*\varepsilon_j^0)}{\sum_{n_j=1}^{B_j-1} a_j^\nu(n_j)\pi_{j,n_j} + \pi_{j,0}}. \quad (26)$$

By Proposition 1, if $\nu \leq \lambda(\mu_j - e^*\varepsilon_j - g)/\mu_j$, for a given g, then $a_j^{\nu,g}(n_j) = 1$, for all $n_j = 1,2,\ldots,B_j-1$; otherwise, $a_j^{\nu,g}(n_j) = 0$. According to our definitions and Corollary 1, $a_j^\nu(n_j) = a_j^{\nu,g^*}(n_j)$, $n_j = 1,2,\ldots,B_j-1$, where $g^* > 0$ is the average reward of process $\mathcal{P}_j^H$ under policy $\phi_j^* \in \Phi_j^H$. Now we split discussion of (26) into two cases. If $\nu \leq \lambda(\mu - e^*\varepsilon_j - g^*)/\mu_j$, then, (26) is equivalent to

$$\nu \leq \frac{\sum_{i=1}^{B_j}\left(\frac{\lambda}{\mu_j}\right)^i(\mu_j - e^*\varepsilon_j + e^*\varepsilon_j^0)}{\sum_{i=0}^{B_j-1}\left(\frac{\lambda}{\mu_j}\right)^i} = \frac{\lambda}{\mu_j}\left(\mu_j - e^*\varepsilon_j + e^*\varepsilon_j^0\right). \quad (27)$$

By definition of our problem, (27) is valid when $\varepsilon_j^0 \geq 0$ and $\nu \leq \lambda(\mu_j - e^*\varepsilon_j + e^*\varepsilon_j^0)/\mu_j$. If $\nu > \lambda(\mu_j - e^*\varepsilon_j - g^*)/\mu_j$, then (26) is equivalent to $\nu \leq \lambda\left(\mu_j - e^*\varepsilon_j + e^*\varepsilon_j^0\right)/\mu_j$. As a consequence,

$$\nu_j^*(0) = \frac{\lambda}{\mu_j}\left(\mu_j - e^*\varepsilon_j + e^*\varepsilon_j^0\right). \quad (28)$$

■

APPENDIX C
PROOF OF PROPOSITION 3

Proof. Based on Proposition 2, the proposition is proved for $n_j = 0$. We assume without loss of generality that $\nu \leq \lambda(\mu_j - e^*\varepsilon_j + e^*\varepsilon_j^0)/\mu_j$, i.e. $a_j^{\nu,g}(0) = 1$.

According to Corollary 1, we obtain

$$g^* = \sum_{n_j \in \mathcal{N}_j} \pi_{j,n_j}\left(R_j(n_j) - a_j^{\nu,g^*}(n_j)\nu\right).$$

Together with Proposition 1, we rewrite $\nu \leq \nu_j^*(n_j,R_j^{g^*})$, $n_j = 1,2,\ldots,B_j-1$ as

$$\nu \leq \frac{\lambda}{\mu_j}\left(\mu_j - e^*\varepsilon_j - (1-\pi_{j,0})(\mu_j - e^*\varepsilon_j) + \pi_{j,0}e^*\varepsilon_j^0 + \nu(1-\pi_{j,B_j})\right)\right) = \frac{\lambda}{\mu_j}(\mu_j - e^*\varepsilon_j + e^*\varepsilon_j^0). \quad (29)$$

As in the case for $\nu \leq \nu^*(n_j,R_j^{g^*})$, $n_j = 1,2,\ldots,B_j-1$, we rewrite $\nu > \nu^*(n_j,R_j^{g^*})$ as

$$\nu > \frac{\lambda}{\mu_j}(\mu_j - e^*\varepsilon_j + e^*\varepsilon_j^0). \quad (30)$$

Equations (29) and (30) prove the proposition. ■

APPENDIX D
CONSEQUENCES OF THE AVERAGING PRINCIPLE

For K_i^0, $i = 1,2,\ldots,\tilde{K}$ with $\sum_{i=1}^{\tilde{K}} K_i^0 = K^0$, define random variables ξ_t as follows. Let $t_{0,k}$, $k = 1,2,\ldots$ and $t_{i,k}$, $i = 1,2,\ldots,K^0$, $k = 1,2,\ldots$, be the times of the kth arrival and of the kth departure from server i, respectively. We assume without loss of generality that $\tilde{K} = K^0$. For the server farm, inter-arrival and inter-departure times are positive with probability one and, also with probability one, two events will not occur at the same time. Define a random vector $\boldsymbol{\xi}_t = (\xi_{0,t},\xi_{1,t},\ldots,\xi_{K^0,t})$ as follows. For $j = 0,1,\ldots,K^0$,

$$\xi_{j,t} = \begin{cases} t_{j,k} - t_{j',k'},\ t_{j,k} = \min\limits_{k''=1,2,\ldots}\{t_{j,k''}|t_{j,k''} > t\}, \\ \qquad t_{j',k'} = \max\limits_{\substack{j''=0,1,\ldots,K^0,\\ k''=1,2,\ldots}}\{t_{j'',k''}|t_{j'',k''} < t_{j,k}\}, \\ \qquad t_{j',k'} \leq t < t_{j,k}, \\ 0,\ \text{otherwise.} \end{cases} \quad (31)$$

Here, the $\boldsymbol{\xi}_t$ is almost surely continuous except for a finite number of discontinuities of the first kind in any bounded interval of $t > 0$.

For $\mathbf{x} \in R^I$, and $i = 1,2,\ldots,I$, we define the action for the Whittle's index policy as

$$a_i^{index}(\mathbf{x}) = \begin{cases} 1 & i = \min\{i|x_i^- > 0,\ i = 1,2,\ldots,I\}, \\ 0 & \text{otherwise,} \end{cases} \quad (32)$$

where $x_i^- = \sum_{k=1}^i x_k$, and the action for the optimal solution of the relaxed problem is

$$a_i^{OPT}(\mathbf{x}) = \begin{cases} 1 & \nu_i^* \geq \nu,\ x_i > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (33)$$

for given state indices ν_i^*, $i \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}$, and ν.

We define a function, $Q^\phi(i,i',\mathbf{x},\boldsymbol{\xi})$, where $\phi \in \Phi$, $i,i' \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}$. For given $\boldsymbol{\xi} = (\xi_0,\xi_1,\ldots,\xi_{K^0}) \in R^{K^0+1}$ and $\mathbf{x} = (x_1,x_2,\ldots,x_I) \in R^I$, $Q^\phi(i,i',\mathbf{x},\boldsymbol{\xi})$ is given, for $i-1,\ i,\ i+1 \in \tilde{\mathcal{N}}_j^{\{0,1\}} \cup \tilde{\mathcal{N}}_j^{\{0,1\}},\ j = 1,2,\ldots,\tilde{K}$ by

$$\begin{aligned} Q^\phi(i,i+1,\mathbf{x},\boldsymbol{\xi}) &= a_i^\phi(\mathbf{x})\tfrac{1}{\xi_0} + f_{i,a}^0(\mathbf{x},\boldsymbol{\xi}), \\ Q^\phi(i,i-1,\mathbf{x},\boldsymbol{\xi}) &= \sum_{j=\lceil x_{i-1}^- \rceil+1}^{\lceil x_i^- \rceil} \tfrac{1}{\xi_j} + f_{i,a}(\mathbf{x},\boldsymbol{\xi}), \\ Q^\phi(i,i',\mathbf{x},\boldsymbol{\xi}) &= 0, \text{ otherwise,} \end{aligned} \tag{34}$$

where ϕ is set to be either $index$ or OPT, $x_i^- = \sum_{k=1}^i x_k$, and, with $0 < a < 1$, $f_{i,a}^0(\mathbf{x},\boldsymbol{\xi})$ and $f_{i,a}(\mathbf{x},\boldsymbol{\xi})$ are appropriate functions to make $Q(i,i',\mathbf{x},\boldsymbol{\xi})$ smooth in $\mathbf{x}$ for all given $\boldsymbol{\xi} \in R^{K^0+1}$ and $0 < a < 1$. Here a is a parameter controlling the Lipschitz constant. Then, for any given $0 < a < 1$, $Q^\phi(i,i',\mathbf{x},\boldsymbol{\xi})$ is bounded and satisfies a Lipschitz condition over any bounded set of $\mathbf{x} \in R^I$ and $\boldsymbol{\xi} \in R^{K^0+1}$.

For $0 < a < 1$ and $\epsilon > 0$, we define $\mathbf{X}_t^{\phi,\epsilon}$ to be a solution of the differential equation

$$\begin{aligned} &\dot{\mathbf{X}}_t^{\phi,\epsilon} = b^\phi(\mathbf{X}_t^\epsilon, \boldsymbol{\xi}_{t/\epsilon}) \\ &= \sum_{i' \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}} Q^\phi(i',i,\mathbf{X}_t^\epsilon,\boldsymbol{\xi}_{t/\epsilon}) - Q^\phi(i,i',\mathbf{X}_t^\epsilon,\boldsymbol{\xi}_{t/\epsilon}), \end{aligned} \tag{35}$$

where ϕ is set to be either $index$ or OPT. It follows that $b^\phi(\mathbf{X}_t^\epsilon,\boldsymbol{\xi}_{t/\epsilon})$ also satisfies a Lipschitz condition on bounded sets in $\mathbf{R}^I \times \mathbf{R}^{K_0+1}$.

From the above definitions, for any $\mathbf{x} \in R^I$, $\delta > 0$, there exists $\overline{b}^\phi(\mathbf{x})$ satisfying

$$\lim_{T\to+\infty} P\left\{\left|\frac{1}{T}\int_t^{t+T} b^\phi(\mathbf{x},\xi_s)\,ds - \overline{b}^\phi(\mathbf{x})\right| > \delta\right\} = 0, \tag{36}$$

uniformly in $t > 0$. Let $\overline{\mathbf{x}}_t^\phi$ be the solution of $\dot{\overline{\mathbf{x}}}_t^\phi = \overline{b}^\phi(\overline{\mathbf{x}}_t)$, $\overline{\mathbf{x}}_0^\phi = \mathbf{X}_0^{\phi,\epsilon}$.

Now we invoke [54, Chapter 7, Theorem 2.1]: if (36) holds, and $\mathbb{E}\left|b^\phi(\mathbf{x},\boldsymbol{\xi}_t)\right|^2 < +\infty$ for all $\mathbf{x} \in R^I$, then, for any $T > 0$, $\delta > 0$,

$$\lim_{\epsilon\to 0} P\left\{\sup_{0\le t\le T}\left|\mathbf{X}_t^{\phi,\epsilon} - \overline{\mathbf{x}}^\phi(t)\right| > \delta\right\} = 0. \tag{37}$$

We scale the time line of the stochastic process by $\epsilon > 0$. As ϵ tends to zero, time is speeded up, and in this way the stochastic process $\{\mathbf{X}_t^{\phi,\epsilon}\}$, driven by the random variable $\boldsymbol{\xi}_{t/\epsilon}$, converges to the deterministic process $\{\overline{\mathbf{x}}^\phi(t)\}$ defined by the differential equation $\overline{b}^\phi(\mathbf{x})$.

Now we interpret the scaling by ϵ in another way. Along similar lines, for a positive integer n and a scaled system, where $K_i = nK_i^0$ replaces K_i^0, $i = 1,2,\ldots,\tilde{K}$, and $K = n\sum_{i=1}^{\tilde{K}} K_i^0$, we define $t_{j,k}^n$,$j = 0,1,\ldots,K$, $k = 1,2,\ldots$ and the random variables $\boldsymbol{\xi}_t^n$ by analogy with the unscaled systems. Then the random variables $\xi_{j,t}^n$, $j = 1,2,\ldots,K$, and $\xi_{0,t}^n$ are exponentially distributed with rate λ_i^0, $i = \lfloor (j-1)/n \rfloor + 1$ and $n\lambda_0^0$, respectively, where λ_i^0, $i = 0,1,\ldots,K^0$, are the corresponding rates for random variables $\xi_{i,t}$. We then define, $Q^{\phi,n}(i,i',\mathbf{x},\boldsymbol{\xi}^n)$ for $\mathbf{x} \in R^I$ and $\boldsymbol{\xi}^n \in R^{nK^0+1}$ as in Equation (34), with appropriate modifications for the change in dimension, where again the functions $f_{i,a}^{0,n}(\mathbf{x},\boldsymbol{\xi}^n)$ and $f_{i,a}^n(\mathbf{x},\boldsymbol{\xi}^n)$ are apprppriately defined to guarantee smoothness of $Q^{\phi,n}(i,i',\mathbf{x},\boldsymbol{\xi})$ $\mathbf{x}$ for all given $\boldsymbol{\xi}^{\boldsymbol{n}} \in R^{nK^0+1}$ and $0 < a < 1$. Here, again, a is a parameter controlling the Lipschitz constant, and ϕ is set to be either $index$ or OPT, $x_i^- = \sum_{k=1}^i x_k$,

In the same vein, for $\mathbf{x} \in R^I$ and $\boldsymbol{\xi}^n \in R^{nK^0+1}$, a differential equation is given by

$$\begin{aligned} &b^{\phi,n}(\mathbf{x},\boldsymbol{\xi}^n) \\ &= \sum_{i' \in \tilde{\mathcal{N}}^{\{0,1\}} \cup \tilde{\mathcal{N}}^{\{0\}}} Q^{\phi,n}(i',i,\mathbf{x},\boldsymbol{\xi}^n) - Q^{\phi,n}(i,i',\mathbf{x},\boldsymbol{\xi}^n). \end{aligned} \tag{38}$$

If we set $\epsilon = 1/n$ then, for any $\mathbf{x} \in R^I$, $n > 0$ and $T > 0$, $\int_0^T b^\phi(\mathbf{x},\xi_{t/\epsilon})dt$ and $\int_0^T (b^{\phi,n}(n\mathbf{x},\xi_t^n)/n)dt$ are equivalently distributed with ϕ set to be either $index$ or OPT. We define $\mathbf{Z}_0^{\phi,\epsilon} = \mathbf{Z}_0^{\phi,n} = \mathbf{x}_0/(K^0+1)$ (there is a zero-reward virtual server), and

$$\dot{\mathbf{Z}}_t^{\phi,n} = \frac{1}{n(K^0+1)} b^{\phi,n}(n(K^0+1)\mathbf{Z}_t^{\phi,n},\xi_t^n),$$

and

$$\dot{\mathbf{Z}}_t^{\phi,\epsilon} = \frac{1}{K^0+1} b^\phi((K^0+1)\mathbf{Z}_t^{\phi,\epsilon},\xi_{t/\epsilon}).$$

From (37), we obtain

$$\lim_{n\to+\infty} P\left\{\sup_{0\le t\le T}\left|\mathbf{Z}_t^{\phi,n} - \overline{\mathbf{x}}^\phi(t)/(K^0+1)\right| > \delta\right\} = 0, \tag{39}$$

for ϕ set to be either $index$ or OPT. Therefore the scaling of time by $\epsilon = 1/n$ is equivalent to scaling of system size by n.

Because of the Lipschitz continuity of $\dot{\mathbf{Z}}_t^{\phi,n}$ and $\dot{\overline{\mathbf{x}}}^\phi(t)$ on $0 < a < 1$, $\lim_{a\to 0} d\dot{\mathbf{Z}}_t^n/da = 0$ and $\lim_{a\to 0} d\dot{\overline{\mathbf{x}}}^\phi(t)/da = 0$, equation (39) holds true in the limiting case as $a \to 0$. Also, if $\mathbf{Z}_0^{\phi,n} = \mathbf{Z}^\phi(0)$, and $\overline{\mathbf{x}}^\phi(0)/(K^0+1) = \mathbf{z}^\phi(0)$, then $\lim_{a\to 0}\overline{\mathbf{x}}^\phi(t)/(K^0+1) \to \mathbf{z}^\phi(t)$ and $\lim_{a\to 0}\mathbf{Z}_t^{\phi,n} \to \mathbf{Z}^\phi(t)$, where ϕ is set to be either $index$ or OPT, $\mathbf{Z}^\phi(t)$ represents the proportions of servers under policy ϕ at time t and $\mathbf{z}^\phi(t)$ is given by (17) (as defined in Section V-D). This leads to (19).

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] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in *Proc. IEEE FOCS*, Milwaukee, WI, Oct. 1995, pp. 374–382.

[5] A. Wierman, L. L. H. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems: Optimality and robustness," *Performance Evaluation*, vol. 69, no. 12, pp. 601–622, Dec. 2012.

[6] S. Albers, F. Müller, and S. Schmelzer, "Speed scaling on parallel processors," *Algorithmica*, vol. 68, no. 2, pp. 404–425, Feb. 2014.

[7] L. Wang, F. Zhang, J. A. Aroca, A. V. Vasilakos, K. Zheng, C. Hou, D. Li, and Z. Liu, "GreenDCN: A general framework for achieving energy efficiency in data center networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 1, pp. 4–15, Jan. 2014.

[8] Y. Tian, C. Lin, and M. Yao, "Modeling and analyzing power management policies in server farms using stochastic Petri nets," in *Proc. e-Energy 2012*. Madrid, Spain: IEEE, May 2012, pp. 1–9.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] D. Niyato, S. Chaisiri, and L. B. Sung, "Optimal power management for server farm to support green computing," in *Proc. IEEE/ACM CCGRID 2009*. Shanghai: IEEE Computer Society, May 2009, pp. 84–91.

[14] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms," in *ACM SIGMETRICS 2009*, Seattle, USA, Jun. 2009, pp. 157–168.

[15] S. Li, S. Wang, T. Abdelzaher, M. Kihl, and A. Robertsson, "Temperature aware power allocation: An optimization framework and case studies," *Sustainable Computing: Informatics and Systems*, vol. 2, no. 3, pp. 117–127, Sep. 2012.

[16] W. Q. M. Guo, A. Wadhawan, L. Huang, and J. T. Dudziak, "Server farm management," Jan. 2014, US Patent 8,626,897. [Online]. Available: http://www.google.com/patents/US8626897

[17] 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*, S. U. Khan and A. Y. Zomaya, Eds. Springer, Mar. 2015, pp. 109–162.

[18] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[19] 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.

[20] 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.

[21] E. Gelenbe and R. Lent, "Energy-QoS trade-offs in mobile service selection," *Future Internet*, vol. 5, no. 2, pp. 128–139, Apr. 2013.

[22] F. Bonomi, "On job assignment for a parallel system of processor sharing queues," *IEEE Trans. Comput.*, vol. 39, no. 7, pp. 858–869, 1990.

[23] V. Gupta, "Stochastic models and analysis for resource management in server farms," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 2011.

[24] 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.

[25] 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.

[26] 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.

[27] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 835–846, Dec. 1997.

[28] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

[29] T. Bonald and A. Proutiere, "Insensitivity in processor-sharing networks," *Performance Evaluation*, vol. 49, no. 1, pp. 193–209, Sep. 2002.

[30] S. Gunawardena and W. Zhuang, "Service response time of elastic data traffic in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 3, pp. 559–570, Mar. 2013.

[31] F. Liu, K. Zheng, W. Xiang, and H. Zhao, "Design and performance analysis of an energy-efficient uplink carrier aggregation scheme," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 2, pp. 197–207, Jan. 2014.

[32] P. Whittle, "Restless bandits: Activity allocation in a changing world," *J. Appl. Probab.*, vol. 25, pp. 287–298, 1988.

[33] 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.

[34] 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.

[35] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano, A. Kanagala, J. Provost, J. Simmons, E. Tanda, J. Wanderer, U. Hölzle, S. Stuart, and A. Vahdat, "Jupiter rising: A decade of Clos topologies and centralized control in Google's datacenter network," in *Proc. ACM SIGCOMM*, London, UK, Aug. 2015, pp. 183–197.

[36] L. A. Barroso, J. Dean, and U. Hölzle, "Web search for a planet: The Google cluster architecture," *IEEE Micro*, vol. 23, no. 2, pp. 22–28, Apr. 2003.

[37] C. Calero, *Handbook of research on web information systems quality*. IGI Global, 2008.

[38] F. A. Haight, "Two queues in parallel," *Biom.*, vol. 45, no. 3-4, pp. 401–410, 1958.

[39] W. Whitt, "Deciding which queue to join: Some counterexamples," *Oper. Res.*, vol. 34, no. 1, pp. 55–62, Jan./Feb. 1986.

[40] Y. Zhao and W. K. Grassmann, "Queueing analysis of a jockeying model," *Oper. Res.*, vol. 43, no. 3, pp. 520–529, May/Jun. 1995.

[41] I. J. B. F. Adan, J. Wessels, and W. H. M. Zijm, "Matrix-geometric analysis of the shortest queue problem with threshold jockeying," *Oper. Res. Lett.*, vol. 13, no. 2, pp. 107–112, Mar. 1993.

[42] E. Hyytiä, 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.

[43] O. T. Akgun, D. G. Down, and R. Righter, "Energy-aware scheduling on heterogeneous processors," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 599–613, Feb. 2014.

[44] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995.

[45] R. R. Weber and G. Weiss, "On an index policy for restless bandits," *J. Appl. Probab.*, no. 3, pp. 637–648, Sep. 1990.

[46] A. Mandelbaum and A. L. Stolyar, "Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule," *Oper. Res.*, vol. 52, no. 6, pp. 836–855, Dec. 2004.

[47] Y. Nazarathy and G. Weiss, "Near optimal control of queueing networks over a finite time horizon," *Ann. Oper. Res.*, vol. 170, no. 1, pp. 233–249, Sep. 2009.

[48] 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.

[49] 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. [Online]. Available: http://projecteuclid.org/euclid.aoap/1472745449

[50] R. Atar and M. Shifrin, "An asymptotic optimality result for the multiclass queue with finite buffers in heavy traffic," *Stoch. Syst.*, vol. 4, no. 2, pp. 556–603, 2014.

[51] M. Larrañaga, U. Ayesta, and I. M. Verloop, "Asymptotically optimal index policies for an abandonment queue with convex holding cost," *Queueing Systems*, pp. 1–71, May 2015.

[52] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 148–177, 1979.

[53] S. M. Ross, *Applied probability models with optimization applications*. Dover Publications (New York), 1992.

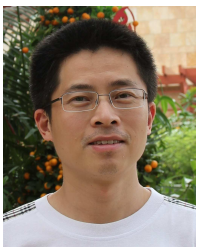
[54] M. I. Freidlin and A. D. Wentzell, *Random perturbations of dynamical systems*. Springer, 2012, vol. 260, translated by J. Szücs.

[55] Standard Performance Evaluation Corporation, tested by Dell Inc. [Online]. Available: https://www.spec.org/power_ssj2008/results/

Jing Fu (S'15-M'16) received the B.Eng. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2011, and the Ph.D. degree in electronic engineering at City University of Hong Kong, Hong Kong, in 2016.

She has been with the School of Mathematics and Statistics, The University of Melbourne, Melbourne, Australia, as a Postdoctoral Research Associate since 2016. Her research interests now include green networking, cloud computing, large-scale multi-queue systems, semi-Markov/Markov decision processes, restless multi-armed bandit problems, asymptotic optimality, fluid control problems.

Bill Moran (M'95) currently serves as a Professor and the Director of Signal Processing and Sensor Control Group in the School of Engineering, RMIT University, Australia. He has been a Professor in the department of Electrical Engineering, University of Melbourne since 2001. Previously he was the Research Director of Defence Science Institute (2011-2014) in University of Melbourne, Professor of Mathematics ('76–'91), Head of the Department of Pure Mathematics ('77–'79, '84–'86), Dean of Mathematical and Computer Sciences ('81, '82, '89) at the University of Adelaide, and Head of the Mathematics Discipline at the Flinders University of South Australia ('91–'95). He was a Chief Investigator ('92–'95), and Head of the Medical Signal Processing Program ('95–'99) in the Cooperative Research Centre for Sensor Signal and information Processing. He was elected to the Fellowship of the Australian Academy of Science in 1984. He holds a Ph.D. in Pure Mathematics from the University of Sheffield, UK ('68), and a First Class Honours B.Sc. in Mathematics from the University of Birmingham ('65). He has been a Principal Investigator on numerous research grants and contracts, in areas spanning pure mathematics to radar development, from both Australian and US Research Funding Agencies, including DARPA, AFOSR, AFRL, Australian Research Council (ARC), Australian Department of Education, Science and Training, DSTO. He is a member of the Australian Research Council College of Experts. His main areas of research interest are in signal processing both theoretically and in applications to radar, waveform design and radar theory, sensor networks, and sensor management. He also works in various areas of mathematics including harmonic analysis, representation theory, and number theory.



Jun Guo (S'01–M'06) received the B.E. degree in automatic control engineering from Shanghai University of Science and Technology, Shanghai, China, in 1992, and the M.E. degree in telecommunications engineering and Ph.D. degree in electrical and electronic engineering from the University of Melbourne, Melbourne, Australia, in 2001 and 2006, respectively.

He was with the School of Computer Science and Engineering, The University of New South Wales, Kensington, Australia, as a Senior Research Associate from 2006 to 2008 and on an Australian Postdoctoral Fellowship supported by the Australian Research Council from 2009 to 2011. From 2012 to 2016, he was with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. He is now an Associate Professor with the College of Computer Science and Technology, Dongguan University of Technology, Dongguan, China. His research is currently focused on green communications and networking, teletraffic theory and its applications in service sectors, and survivable network topology design.

Eric W. M. Wong (S'87–M'90–SM'00) received the B.Sc. and M.Phil. degrees in electronic engineering from the Chinese University of Hong Kong, Hong Kong, in 1988 and 1990, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Massachusetts, Amherst, MA, USA, in 1994.

He is an Associate Professor with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. His research interests include analysis and design of telecommunications and computer networks, energy-efficient data center design, green cellular networks and optical switching.

Moshe Zukerman (M'87–SM'91–F'07) received the B.Sc. degree in industrial engineering and management and the M.Sc. degree in operations research from the Technion—Israel Institute of Technology, Haifa, Israel, in 1976 and 1979, respectively, and the Ph.D. degree in engineering from the University of California, Los Angeles, CA, USA, in 1985.

He was an independent consultant with the IRI Corporation and a Postdoctoral Fellow with the University of California, Los Angeles, from 1985 to 1986. He was with the Telstra Research Laboratories (TRL), Melbourne, Australia, first as a Research Engineer from 1986 to 1988, and as a Project Leader from 1988 to 1997. He also taught and supervised graduate students with Monash University, Melbourne, Australia, from 1990 to 2001. From 1997 to 2008, he was with The University of Melbourne, Melbourne, Australia. In 2008, he joined City University of Hong Kong, Hong Kong, as a Chair Professor of Information Engineering and a team leader.

Prof. Zukerman has served on various editorial boards such as Computer Networks, IEEE Communications Magazine, IEEE Journal of Selected Areas in Communications, IEEE/ACM Transactions on Networking, and the International Journal of Communication Systems.