Near-Optimal Stochastic Bin-Packing in Large Service Systems with Time-Varying Item Sizes

YIGE HONG, Carnegie Mellon University, USA
QIAOMIN XIE, University of Wisconsin-Madison, USA
WEINA WANG, Carnegie Mellon University, USA

In modern computing systems, jobs' resource requirements often vary over time. Accounting for this temporal variability during job scheduling is essential for meeting performance goals. However, theoretical understanding on how to schedule jobs with time-varying resource requirements is limited. Motivated by this gap, we propose a *new setting* of the stochastic bin-packing problem in service systems that allows for *time-varying* job resource requirements, also referred to as 'item sizes' in traditional bin-packing terms. In this setting, a job or 'item' must be dispatched to a server or 'bin' upon arrival. Its resource requirement may vary over time while in service, following a Markovian assumption. Once the job's service is complete, it departs from the system. Our goal is to minimize the expected number of active servers, or 'non-empty bins', in steady state.

Under our problem formulation, we develop a job dispatch policy, named Join-Requesting-Server (JRS). Broadly, JRS lets each server independently evaluate its current job configuration and decide whether to accept additional jobs, balancing the competing objectives of maximizing throughput and minimizing the risk of resource capacity overruns. The JRS dispatcher then utilizes these individual evaluations to decide which server to dispatch each arriving job to. The theoretical performance guarantee of JRS is in the asymptotic regime where the job arrival rate scales large linearly with respect to a scaling factor r. We show that JRS achieves an additive optimality gap of $O(\sqrt{r})$ in the objective value, where the optimal objective value is $\Theta(r)$. When specialized to constant job resource requirements, our result improves upon the state-of-the-art $o(r)$ optimality gap. Our technical approach highlights a novel policy conversion framework that reduces the policy design problem into a single-server problem.

CCS Concepts: • **Networks** → **Network performance analysis**; • **Mathematics of computing** → **Markov processes**.

Additional Key Words and Phrases: stochastic bin-packing, large service systems, policy conversion, asymptotic optimality

ACM Reference Format:
Yige Hong, Qiaomin Xie, and Weina Wang. 2023. Near-Optimal Stochastic Bin-Packing in Large Service Systems with Time-Varying Item Sizes. *Proc. ACM Meas. Anal. Comput. Syst.* 7, 3, Article 48 (December 2023), 46 pages. https://doi.org/10.1145/3626779

1 INTRODUCTION

1.1 Background and Motivation

In modern computing systems, a job often takes the form of a virtual machine (VM) or a container [8, 14]. Such a job comes with a resource requirement, such as a certain number of CPUs and amount of memory, while in service. Each server in the system offers a limited amount of these resources.

Authors' addresses: Yige Hong, yigeh@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 15213; Qiaomin Xie, qiaomin.xie@wisc.edu, University of Wisconsin-Madison, Madison, Wisconsin, USA, 53706; Weina Wang, weinaw@cs.cmu.edu, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 15213.

2476-1249/2023/12-ART48
https://doi.org/10.1145/3626779

When a job arrives at the system, the job dispatch policy needs to decide which server the job should be assigned to, given the job's resource requirement and servers' current job configurations. This job scheduling problem can be approached as a *Stochastic Bin-Packing (SBP) problem*, where jobs are viewed as items, job resource requirements as item sizes, and servers as bins. A traditional SBP setting considers a finite set of jobs that arrive online but do not depart from the system. The objective is to minimize the number of servers that have jobs on them, or 'non-empty bins', subject to the resource capacities of the servers. SBP, with a rich history in operations research and theoretical computer science [10–12], is a field of continuous developments and advancements [1, 16, 18].

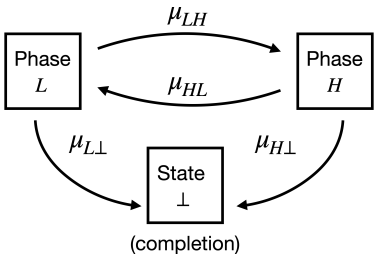
To incorporate job *departures* into the problem formulation, a setting referred to as *stochastic bin-packing in service systems* has been proposed recently [17, 32–36]. In this setting, jobs not only arrive but also depart over time. More specifically, jobs are assumed to arrive according to Poisson processes, and each job is assumed to stay in the system for an exponentially distributed service time. The service time of a job remains unknown until the job departs. Before delving further into SBP in service systems, it is worth mentioning that there is a parallel thread of research on the so-called dynamic bin-packing problem that also handles job departures (see, e.g., [6, 9, 20], and references therein), but it is primarily from a worst-case analysis perspective. Additionally, the virtual machine scheduling problem with objectives different from minimizing the number of active servers has also been widely studied (see, e.g., [22–24, 26–29, 39]).

For SBP in service systems, the goal is to design a job dispatch policy σ that minimizes the expected number of active servers in *steady state*, denoted as $N(\sigma)$. The *optimality gap* of a policy σ is defined as $N(\sigma) - N(\sigma^*)$, where σ^* is the optimal policy. Since SBP in service systems aims to model today's large-scale computing systems, the optimality gap of a policy is usually studied in the regime where the total job arrival rate becomes large. As we scale up the total job arrival rate linearly with a scaling factor, r, the optimal value $N(\sigma^*)$ can be shown to be $\Theta(r)$. [1] Therefore, we say a policy is asymptotically optimal if its optimality gap is $o(r)$.

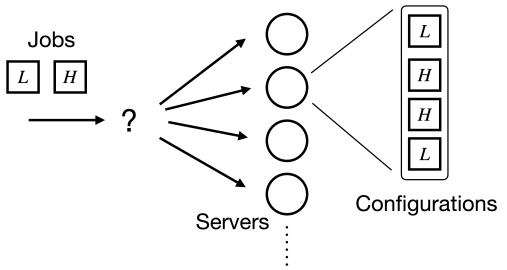
The optimality gap for SBP in service systems has been characterized in the line of work [32–36]. In particular, Stolyar [32], Stolyar and Zhong [34] propose greedy policies that are asymptotically optimal, but the scheduler that executes these policies needs to know detailed state information, which is in a high-dimensional space. Stolyar and Zhong [35, 36] later develop policies that use much less state information and achieve $\Theta(r)$ (with an arbitrarily small constant) and $o(r)$ optimality gaps, respectively.

While prior work on SBP in service systems has provided substantial insights into scheduling virtual-machine-type jobs, it primarily focuses on job resource requirements that remain fixed over time. However, in modern computing systems, jobs' resource requirements often vary over time [2, 13, 21, 30, 31, 37]. For example, when a job involves providing user-facing services, the instantaneous requirement on CPUs and memory depends on the service demand, which is subject to fluctuation over time [13, 21]. Time-varying job resource requirements pose significant challenges in optimizing system efficiency, particularly when aiming to minimize the number of active servers, thereby improving server utilization. It is pertinent to note that low utilization has been recognized as a significant obstacle to the continued scaling of today's computing systems.

Motivated by this gap, in this paper, we propose a *new setting of stochastic bin-packing in service systems* that allows job resource requirements, or 'item sizes', to vary over time.



(a) A simplified version of our job model. Each job in service is in either an L phase or an H phase, associated with low and high resource requirements, respectively. When the job is completed, it is said to be in the state $\perp$. The job transitions between the two phases while in service until it is completed, following a continuous-time Markov chain with rates $\mu_{ii'}$, $i, i' \in \{L, H, \perp\}$.



(b) A system model with an infinite number of identical servers. As soon as a job arrives to the system, the job needs to be dispatched to a server to start service immediately. The configuration of each server is the number of jobs in each phase on the server.

Fig. 1. Job model and system model.

1.2 Problem Formulation: A Simplified Version

We first describe our job model that features time-varying resource requirements. For ease of exposition, here we present a simplified setting where each job in service can be in one of the two *phases*, L and H, associated with *low* and *high* resource requirements, respectively. Our full model, presented in Section 2, allows *more than two phases and more than one type of resources*. To model the temporal variation in the resource requirement, we assume that each job transitions between the two phases while in service until it is completed, following a continuous-time Markov chain illustrated in Figure 1(a). We use an absorbing state $\perp$ to denote that the job is completed. A job can initialize in either phase L or phase H, and they are referred to as *type L* and *type H* jobs, respectively. Note that the setting where a job's resource requirement does not vary over time is a special case of our job model where the transition rates between phases are 0.

We consider a system with an infinite number of identical servers, illustrated in Figure 1(b). We assume jobs arrive according to a Poisson process as existing work on SBP in service systems. In particular, we assume that the two types of jobs arrive at the system following two independent Poisson processes, with rates Λ_L and Λ_H, respectively; i.e., the interarrival times of type L and type H jobs are i.i.d. following exponential distributions with means $1/\Lambda_L$ and $1/\Lambda_H$, respectively. Upon arrival, a job needs to be dispatched to a server according to a *dispatch policy*, and the job enters service immediately. The goal is to design a policy σ to minimize the expected number of *active servers* (servers currently serving a positive number of jobs) in steady state, denoted as $N(\sigma)$.

As job resource requirements vary over time, situations can arise where the total job resource requirement on a server exceeds the server's resource capacity, resulting in resource contention. Modern computing systems can tolerate temporary overruns of resource capacity, though they often incur performance degradation or other costs [7, 15]. In our model, we incorporate a rate at which the cost accumulates due to resource contention. We first represent the state of a server by its *configuration*, a vector $\boldsymbol{k} = (k_L, k_H)$ where k_L and k_H are the numbers of jobs in phase L and phase H, respectively. Then a *cost rate function* $h(\cdot)$ maps a server's configuration to a rate of cost. For example, the cost rate can be proportional to how much the total resource requirement of

[1]We use the standard Bachmann–Landau notation. Consider two functions $a(r)$ and $b(r)$, where $b(r)$ is positive for large enough r. Then $a = O(b)$ if $\limsup_{r\to+\infty} \frac{|a(r)|}{b(r)} < \infty$; $a = o(b)$ if $\lim_{r\to+\infty} \frac{a(r)}{b(r)} = 0$; $a = \Theta(b)$ if $a = O(b)$ and $b = O(a)$.

the jobs on the server exceeds this server's resource capacity. A more general definition of $h(\cdot)$ is given in Section 2. We assume that the resource contention does not affect the transition rates in the job model nor prompt jobs to be terminated, suitable for the application scenarios where the contention level is low and manageable. Let $C(\sigma)$ denote the average expected cost rate per server.

Now our bin-packing problem can be formulated as follows:

$$\begin{aligned} \underset{\sigma}{\text{minimize}} \quad & N(\sigma) \\ \text{subject to} \quad & C(\sigma) \le \epsilon, \end{aligned} \tag{1}$$

where $\epsilon > 0$ is a budget for the cost rate of resource contention. We are interested in solving this problem in the asymptotic regime where the arrival rates (Λ_L, Λ_H) scale to infinity [33–36], motivated by the ever-increasing computing demand that drives today's computing systems to be large-scale. Specifically, we assume $(\Lambda_L, \Lambda_H) = (\lambda_L r, \lambda_H r)$ for some fixed coefficients λ_L and λ_H and a scaling factor r, and we study the asymptotic regime where r increases.

1.3 Main Result

Our main result is an *asymptotically optimal* policy, named Join-Requesting-Server (JRS), for this new setting of SBP in service systems with time-varying job resource requirements. The asymptotic optimality is in the sense that under our proposed policy JRS, the expected number of active servers is at most $\left(1 + O\left(r^{-0.5}\right)\right)$ times the optimal objective value of the optimization problem in (1), while the cost rate incurred is at most $\left(1 + O\left(r^{-0.5}\right)\right) \cdot \epsilon$ (i.e., exceeding the budget by at most a diminishing fraction). This asymptotic optimality result translates into an *additive optimality gap* of $O(\sqrt{r})$ in the objective value (expected number of active servers), since the optimal objective value can be shown to be $\Theta(r)$. This main result is formally presented in Theorem 1.

Our model can be specialized to the traditional setting of SBP in service systems where jobs' resource requirements remain fixed over time. For this specialization, we replace the constraint $C(\sigma) \le \epsilon$ in the problem formulation (1) with a capacity constraint, which requires the total resource requirement by jobs on a server to be within the server's resource capacity. Our proposed policy JRS can then be adapted into one that has an $O(\sqrt{r})$ optimality gap in the objective value, which improves upon the state-of-the-art $o(r)$ optimality gap. A discussion on the implementation complexity of JRS and how it compares with existing policies for the traditional setting of SBP in service systems is provided in Section 4.3. To be clear, this setting is not a strict special case of the formulation in (1) because $\epsilon > 0$ is required there, but our approach and proof carry over.

From a technical approach perspective, our contribution is a novel approach that decomposes the policy design into two steps: defining a single-server sub-problem, and then converting the solution of the sub-problem into a policy in the original problem. This decomposition not only reduces the complexity of policy design but also makes the analysis tractable. We provide an overview of this approach in Section 1.4.

1.4 Approach Overview

To motivate our approach, we ask two questions:

How should we design a good dispatch policy for this system?
How can we prove that a dispatch policy is asymptotically optimal?

Before presenting our answers to these two questions, we first comment on why they are challenging to answer. On the one hand, solving this problem directly via dynamic programming is intractable due to the unbounded state space resulting from the infinitely many servers. Even if we restrict ourselves to the servers that are active, the state space is still prohibitively large. On the other hand, we can consider designing a heuristic policy. However, unlike traditional SBP problems where we

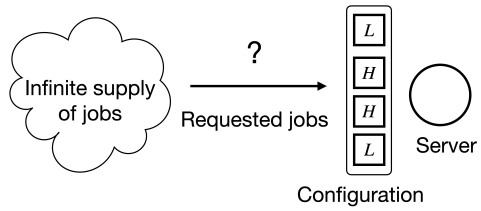


Fig. 2. A single-server system with an infinite supply of jobs. A single-server policy decides when to request jobs and how many jobs of each type to request.

can simply seek to pack the servers as compactly as possible, here the question of *how many jobs should be put on a server* is complicated. The complexity comes from the time-varying resource requirements of jobs, which affect future resource contention.

A policy-conversion framework. We answer the two questions at the same time with a novel *policy-conversion* framework. The framework has two steps:

(1) Define the *single-server problem*, which is a easy-to-solve low-dimensional subproblem;
(2) Convert the optimal policy of the single-server problem into a policy in the original problem.

This framework allows us to break down the complicated policy design problem into two components. In defining the single-server problem in Step 1, our goal is to quantify the throughput of each server under the resource contention constraint; in the policy conversion in Step 2, our goal is to dispatch jobs optimally based on each single server's characteristics. As we will show, a careful construction of the single-server problem and the conversion procedure naturally leads to a policy for the original system and a proof of its asymptotic optimality. Below we give a quick overview of how we carry out these two steps and the motivation for the design choices.

Single-server problem. To define the single-server problem, consider the following setting: suppose that our goal is to maximize the throughput of *one specific server* while keeping its expected cost rate of resource contention below ϵ, then how should we send jobs to this server? Observe that even though we want to send jobs to the server as frequently as possible, the frequency is fundamentally limited by how fast the server is able to serve jobs and how many jobs can be packed on the server. This motivates us to consider the single-server system illustrated in Figure 2. The system has one server and an infinite supply of jobs of all types, so the server can start the service of any number of new jobs of any type at any time. We say the server *requests* a job from the infinite supply whenever it starts serving a new job. We assume the same job model and cost model as in the original infinite-server system. The single-server problem aims to find a job-requesting policy that maximizes the throughput (the number of each type of job that can be served) along the direction of the arrival rate vector $(\Lambda_L, \Lambda_H) = (\lambda_L r, \lambda_H r)$, while maintaining the steady-state expected cost rate of resource contention below ϵ.

How is the single-server problem related to the original problem? Let $\overline{N}^*$ be the number such that the total throughput of $\overline{N}^*$ single-server systems under the optimal job-requesting policy is equal to $(\lambda_L r, \lambda_H r)$ (assuming $\overline{N}^*$ is an integer for simplicity). Consider the following policy in the original system: let each of the first $\overline{N}^*$ servers in the original system adopt the optimal job-requesting policy and send requests to the dispatcher based on its current configuration. If the requested jobs were to arrive as soon as the dispatcher received the requests, the dispatcher would be able to fulfill the requests immediately. In this case, the first $\overline{N}^*$ servers in the original system would have the same dynamics as $\overline{N}^*$ i.i.d. single-server systems, achieving the largest possible throughput and

satisfying the constraint on resources contention. So the original system would have achieved the optimal number of active servers.

However, in the actual model, the dispatcher cannot immediately fulfill a job request because jobs arrive stochastically over time. Nevertheless, the dispatcher can still find a suitable way to match each job arrival with the requests. To see this, note that the time points when the dispatcher receives type i requests result from the superposition of $\overline{N}^* = \Theta(r)$ independent point processes, each with the average rate $\lambda_i r/\overline{N}^*$. As $r \to \infty$, the instantaneous rates of requesting type i jobs concentrate around the arrival rate $\lambda_i r$ for each i. As a result, most job requests can be fulfilled within a diminishing delay, so most servers in the original system can closely track the optimal single-server dynamics.

A meta-policy, Join-Requesting-Server (JRS), and its asymptotic optimality. Based on the single-server problem and the idea of tracking the optimal single-server dynamics, we propose a meta-policy, Join-Requesting-Server (JRS), which converts a single-server policy $\overline{\sigma}$ to a dispatch policy in the original infinite-server system. We say that JRS takes $\overline{\sigma}$ as a subroutine. The full definition of JRS is given in Section 4, along with discussions on various practical considerations in its implementation.

We show that the asymptotic performance of JRS is related to its subroutine in the sense described in Theorem 3, which we refer to as the *conversion theorem*. In particular, JRS with the optimal single-server policy (which we refer to as Single-OPT) as the subroutine is asymptotically optimal for the original infinite-server problem as $r \to \infty$.

In order to track the optimal single-server dynamics, JRS uses a more sophisticated mechanism to control the long-term consequences of missing job requests or fulfilling requests with delays. The mechanism involves the auxiliary variables of *tokens* and *virtual jobs*, which regularize the process of generating requests and matching arrivals with requests. These auxiliary variables play a crucial role in the proof of Theorem 3 in Section 5, where a novel Stein's method argument is carried out. We discuss the role of tokens and virtual jobs and their necessity at the end of Section 4 and in Section 5.5.

Finally, we comment that our policy conversion framework can be applicable to other systems with similar structures. Specifically, we can try to define a suitable single-server problem, solve for its optimal policy, and convert the optimal single-server policy to the original problem. A similar conversion theorem should hold as long as the servers are weakly coupled in some sense. See Section 6 for a discussion of such systems.

Relation to the mean-field approach. We remark that the mean-field approach often studies the empirical distribution of configurations on all servers [17, 32–36], which can be viewed as a probability distribution of a single server's configuration. However, the mean-field approach is typically used to *analyze* this empirical distribution under a given policy for the original system. In contrast, our approach solves the single-server problem to *design* a single-server policy, and then converts it to a policy in the original system with performance guarantee.

1.5 Paper organization

In Section 2, we present the general problem formulation, which generalizes the simplified version in this section. In Section 3, we give a more detailed overview of our main result and approach, with a short proof of Theorem 1 (main result) based on Theorems 2–4 at the end of the section. Section 4 provides a detailed description of our meta-policy, Join-Requesting-Server (JRS), along with discussions on practical considerations in its implementation. In Section 5, we prove the performance guarantee of JRS (Theorem 3) under an irreducibility assumption. The proof for the

general case and other proofs are deferred to the appendices. We conclude the paper and discuss some future directions in Section 6.

2 PROBLEM FORMULATION

Job Model. As described in Section 1, we consider a job model where each job in service can be in one of multiple *phases*, each phase associated with a different resource requirement. Here the resource requirement can be a multi-dimensional vector, with each coordinate specifying the requirement of one type of resource. To model the temporal variation in the resource requirement, we assume that each job transitions between phases while in service until it is completed. The phase transition process is described by a continuous-time Markov chain on the state space $\mathcal{I} \cup \{\perp\}$, where $\mathcal{I}$ is the set of phases and $\perp$ is the absorbing state that denotes the completion of the job. We call a transition between two phases in $\mathcal{I}$ an internal transition, and let $\mu_{ii'}$ denote the transition rate from phase i to phase i'; the departure of a job then corresponds to a transition from a phase $i \in \mathcal{I}$ to $\perp$, whose transition rate is denoted as $\mu_{i\perp}$. The phase transitions of different jobs are assumed to be independent of each other.

We classify a job as a type i job if it starts from phase $i \in \mathcal{I}$ when entering service. Jobs of each type i arrive to the system according to an independent Poisson process with rate Λ_i.

Server Model. We consider an infinite-server system with identical servers. As soon as a job arrives to the system, the job needs to be dispatched to a server to start service immediately. Note that this is always feasible because there are an infinite number of servers in the system. We assume that jobs cannot be preempted or migrated. To describe the state of a server, we define the *configuration* of a server as an $|\mathcal{I}|$-dimensional vector $\boldsymbol{k} = (k_i)_{i\in\mathcal{I}}$, whose i-th entry k_i is the number of jobs in phase i on the server. Each server has a limit on the total number of jobs in service at the same time. This limit is denoted as $K_{\max}$ and referred to as the *service limit*. Then the set of feasible server configurations is $\mathcal{K} \triangleq \{\boldsymbol{k} \colon \sum_{i\in\mathcal{I}} k_i \le K_{\max}\}$.

System Dynamics. The system state can be represented by the concatenation of the configurations of all servers. Specifically, we index the servers by positive integer numbers and denote the configuration of server ℓ at time t as $\boldsymbol{K}^\ell(t)$. Then the state of the entire system can be represented by the infinite vector $(\boldsymbol{K}^\ell(t))_{\ell\in\mathbb{Z}_+}$.

Suppose that the system is in state $(\boldsymbol{k}^\ell)_{\ell\in\mathbb{Z}_+}$. Let $\boldsymbol{e}_i$ be an $|\mathcal{I}|$-dimensional vector whose i-th entry is 1 and all other entries are 0. Then the following state transitions can happen:

- $\boldsymbol{k}^\ell \to \boldsymbol{k}^\ell + \boldsymbol{e}_i, \boldsymbol{k}^{\ell'} \to \boldsymbol{k}^{\ell'}\ \forall \ell' \ne \ell$: a type i job arrives and is dispatched to server ℓ;
- $\boldsymbol{k}^\ell \to \boldsymbol{k}^\ell + \boldsymbol{e}_{i'} - \boldsymbol{e}_i, \boldsymbol{k}^{\ell'} \to \boldsymbol{k}^{\ell'}\ \forall \ell' \ne \ell$: a job on server ℓ transits from phase i to phase i';
- $\boldsymbol{k}^\ell \to \boldsymbol{k}^\ell - \boldsymbol{e}_i, \boldsymbol{k}^{\ell'} \to \boldsymbol{k}^{\ell'}\ \forall \ell' \ne \ell$: a job on server ℓ departs the system from phase i.

The specifics of the system dynamics depend on the employed *dispatch policy* that decides which server to dispatch to when a job arrives.

Active Servers. We are interested in the number of *active servers*, i.e., servers currently serving a positive number of jobs. Note that given the arrival rates of jobs, the smaller the number of active servers, the better the system is utilized. Let $X_{\boldsymbol{k}}(t)$ be the number of servers in configuration $\boldsymbol{k}$ at time t, i.e., $X_{\boldsymbol{k}}(t) = \sum_{\ell=1}^{\infty} \mathbb{1}_{\{\boldsymbol{K}^\ell(t)=\boldsymbol{k}\}}$. Then the number of active servers can be written as $\sum_{\boldsymbol{k}\ne\boldsymbol{0}} X_{\boldsymbol{k}}(t)$, where $\boldsymbol{0} \in \mathbb{R}^{|\mathcal{I}|}$ is the zero vector.

Cost of Resource Contention. Recall that the cost rate function $h(\cdot)$ maps a server's configuration to a rate of cost. We assume that $h(\cdot)$ is any function that is Γ-Lipschitz continuous with respect to the L^1 distance for some constant $\Gamma > 0$ and satisfies $h(\boldsymbol{0}) = 0$.

Performance Goal. Our high-level goal is to design dispatch policies that minimize the number of active users while keeping the cost rate of resource contention within a certain budget. Specifically, we consider policies that are allowed to be randomized and non-Markovian (i.e., the policies can make history-dependent decisions). We further focus on policies that induce a unique stationary distribution on the configuration process $\{(\boldsymbol{K}^\ell(t))_{\ell\in\mathbb{Z}_+}\}$, assuming that the configuration process is embedded in a Markov chain that has a unique stationary distribution. We are interested in such policies because the resulting time averages of quantities related to the configurations are equal to the corresponding expectations under the unique stationary distribution regardless of the initial state. Let σ be a policy of interest, $(\boldsymbol{K}^\ell)_{\ell\in\mathbb{Z}_+}$ be a random element that follows the stationary distribution of the system state induced by σ, and $X_{\boldsymbol{k}}$ be the corresponding number of servers in configuration $\boldsymbol{k}$ in steady state under σ. Then the expected number of active servers is given by

$$N(\sigma) \triangleq \sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}].$$

We define the expected cost rate per expected active server as

$$C(\sigma) \triangleq \frac{\sum_{\boldsymbol{k}\neq\boldsymbol{0}} h(\boldsymbol{k})\mathbb{E}[X_{\boldsymbol{k}}]}{\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}]}.$$

Note that if $C(\sigma) \le \epsilon$, we have $\sum_{\boldsymbol{k}\neq\boldsymbol{0}} h(\boldsymbol{k})\mathbb{E}[X_{\boldsymbol{k}}] \le \epsilon \sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}]$. Now our goal can be formulated as the following optimization problem, referred to as problem $\mathcal{P}((\Lambda_i)_{i\in\mathcal{I}}, \epsilon)$:

$$\begin{aligned} \underset{\sigma}{\text{minimize}} \quad & N(\sigma) \\ \text{subject to} \quad & C(\sigma) \le \epsilon, \end{aligned} \tag{2}$$

where ϵ is a budget for the cost rate of resource contention.

Asymptotic Optimality. We focus on the asymptotic regime where for all $i \in \mathcal{I}$, the arrival rate is given by $\Lambda_i = \lambda_i r$ for some constant coefficient λ_i and a positive *scaling factor* $r \to +\infty$. To define asymptotic optimality, we first define the following notion of approximation to the optimization problem $\mathcal{P}((\Lambda_i)_{i\in\mathcal{I}}, \epsilon)$ in (2): a policy σ is said to be (α, β)-optimal if $N(\sigma) \le \alpha \cdot N^*((\Lambda_i)_{i\in\mathcal{I}}, \epsilon)$ and $C(\sigma) \le \beta \cdot \epsilon$, where $N^*((\Lambda_i)_{i\in\mathcal{I}}, \epsilon)$ is the optimal objective value in (2). Now consider a family of policies $\sigma^{(r)}$ indexed by the scaling factor r. We say that the policy $\sigma^{(r)}$ is *asymptotically optimal* if it is $\left(\alpha^{(r)}, \beta^{(r)}\right)$-optimal to the optimization problem $\mathcal{P}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ with $\alpha^{(r)}, \beta^{(r)} \to 1$ as $r \to \infty$. We will suppress the superscript (r) for simplicity when it is clear from the context.

We note that under any policy σ, $N(\sigma) = \Theta(r)$. This can be proven using the renowned Little's Law [19] in the following way. The total job arrival rate is $\Theta(r)$ and the expected time that a job spends in the system is $O(1)$. So by Little's Law, the expected number of jobs in the system in steady state is $\Theta(r)$. Since each server can accommodate a constant number of jobs, the expected number of active servers is $\Theta(r)$. Given this, $\left(\alpha^{(r)}, \beta^{(r)}\right)$-optimality implies an optimality gap of $\alpha^{(r)} \cdot r$ in the objective value.

3 MAIN RESULT AND OUR APPROACH

3.1 Main Result

Our main result, Theorem 1, is the asymptotic optimality of our proposed policy Join-Requesting-Server (JRS), with a subroutine we call Single-OPT, as briefly discussed in Section 1. This asymptotic optimality result implies an $O(\sqrt{r})$ optimality gap in the expected number of active servers. We defer the detailed descriptions of JRS and Single-OPT to Section 4 and Appendix C. Theorem 1 follows immediately from Theorems 2–4 to be introduced in Section 3.2; a short proof is included at the end of Section 3.2 for clarity.

Theorem 1 (Asymptotic Optimality). *Consider a stochastic bin-packing problem in service systems with time-varying job resource requirements. Let the arrival rates be $(\lambda_i r)_{i\in\mathcal{I}}$ and the cost rate budget be $\epsilon > 0$. Then the policy Join-Requesting-Server (JRS) with the subroutine Single-OPT is $\left(1 + O\left(r^{-0.5}\right), 1 + O\left(r^{-0.5}\right)\right)$-optimal. That is, the expected number of active servers under JRS with Single-OPT is at most $\left(1 + O\left(r^{-0.5}\right)\right)$ times the optimal value of the problem $\mathcal{P}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$, while the cost rate incurred is at most $\left(1 + O\left(r^{-0.5}\right)\right) \cdot \epsilon$.*

Specialization to Non-Time-Varying Resource Requirements. As mentioned in Section 1, we can specialize this result to the setting where the resource requirement of a job does not vary over time. To do that, we remove the cost constraint in $\mathcal{P}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$, and redefine the set of feasible server configurations, $\mathcal{K}$, to also incorporate hard capacity constraints for each type of resources. The rest of the analysis is almost identical to that of the analysis for time-varying resource requirements; we omit the details due to the space limit. This specialization results in a policy that is $\left(1 + O\left(r^{-0.5}\right)\right)$-optimal in the expected number of active servers.

3.2 Our Approach

In a nutshell, our approach is to reduce the original optimization problem in an infinite-server system to an optimization problem in a single-server system, which is defined below.

A Single-Server System. Consider a single-server system serving jobs with time-varying resource requirements. The system has *an infinite supply of jobs* of all types. As a result, the server can *request* any number of new jobs of any type at any time. Once a job is requested, it immediately enters service.

We represent the server configuration at time t using a vector $\overline{\boldsymbol{K}}(t) = (\overline{K}_i(t))_{i\in\mathcal{I}}$, whose i-th entry denotes the number of jobs in phase i. We assume that the single-server system has the same service limit $K_{\max}$ and cost rate function $h(\cdot)$ as a server in the original infinite-server system. Therefore, the server configuration $\overline{\boldsymbol{K}}(t)$ is also in the set $\mathcal{K} = \{\boldsymbol{k} \colon \sum_{i\in\mathcal{I}} k_i \le K_{\max}\}$, and the cost rate at time t is $h(\overline{\boldsymbol{K}}(t))$.

A single-server policy $\overline{\sigma}$ determines when and how many jobs of each type to request. We allow the single-server policy to be randomized and assume it is Markovian, i.e., it makes decisions only based on the current configuration. Note that allowing non-Markovian policies will not change the optimal value of the single-server problem that we will consider (see Appendix C). Let $\pi \triangleq (\pi(\boldsymbol{k}))_{\boldsymbol{k}\in\mathcal{K}}$ be a stationary distribution of the server configuration under the policy $\overline{\sigma}$, and let $\overline{\boldsymbol{K}}(\infty)$ be a random variable with the distribution π. When we consider a policy $\overline{\sigma}$ and its stationary distribution π, we assume that the system is initialized from π. The policy $\overline{\sigma}$ together with π defines the *request rate* of type i jobs $\overline{\lambda}_i$, which is the expected number of type i jobs requested per unit time in steady state. Note that $\overline{\lambda}_i$ is the throughput of type i jobs since the system has a finite state space.

We consider the following single-server problem, denoted as $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$:

$$\begin{aligned} \underset{\overline{N},\ \overline{\sigma},\ \pi}{\text{minimize}} \quad & \overline{N} \\ \text{subject to} \quad & \mathbb{E}\left[h\left(\overline{\boldsymbol{K}}(\infty)\right) \middle| \overline{\boldsymbol{K}}(\infty) \neq \boldsymbol{0}\right] \le \epsilon, \\ & \overline{N}\,\overline{\lambda}_i = \lambda_i r, \quad \forall i \in \mathcal{I}. \end{aligned} \tag{3}$$

The single-server problem can be interpreted as follows. We can think of $\overline{N}$ as the number of copies of the single-server system under $\overline{\sigma}$ needed to support the arrival rates $(\lambda_i r)_{i\in\mathcal{I}}$ in the infinite-server system. To minimize $\overline{N}$, it is equivalent to maximizing the throughput $(\overline{\lambda}_i)_{i\in\mathcal{I}}$ in each single-server system, while maintaining their proportions as $(\lambda_i r)_{i\in\mathcal{I}}$.

We remark that for the problem $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$, we only need to consider policies that do not depend on the scaling factor r. To see this, we can replace the decision variable $\overline{N}$ with $\overline{n} \triangleq \overline{N}/r$ and the optimization problem can be equivalently formulated as follows, which does not involve r:

$$\begin{aligned} \underset{\overline{n},\ \overline{\sigma},\ \pi}{\text{minimize}} \quad & \overline{n} \\ \text{subject to} \quad & \mathbb{E}\left[h\big(\overline{\boldsymbol{K}}(\infty)\big)\middle|\overline{\boldsymbol{K}}(\infty) \neq \boldsymbol{0}\right] \leq \epsilon, \\ & \overline{n}\,\overline{\lambda}_i = \lambda_i, \quad \forall i \in \mathcal{I}. \end{aligned} \tag{4}$$

Lower Bound. The single-server problem gives a lower bound to the original problem in (2) as stated in the following theorem. The proof is given in Appendix A.

Theorem 2 (Lower Bound)**.** *Consider a stochastic bin-packing problem in service systems with time-varying job resource requirements. Let the arrival rates be $(\lambda_i r)_{i\in\mathcal{I}}$ and the cost rate budget be $\epsilon > 0$. Let N^* be the optimal value of the original infinite-server problem in (2), and let $\overline{N}^*$ be the optimal value of the single-server problem $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$, then $N^* \geq \overline{N}^*$.*

Converting From the Single-Server System to the Infinite-Server System. Having established a lower bound on the infinite-server problem $\mathcal{P}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ in terms of the optimal value of the single-server problem $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$, next we focus on finding an asymptotically optimal policy. We will characterize the performance guarantee of a class of policies and then show that the best policy within the class is asymptotically optimal. Specifically, we consider a meta-policy called Join-Requesting-Server (JRS), which converts a Markovian single-server policy $\overline{\sigma}$ into an infinite-server policy. We call the policy resulting from the conversion *a JRS policy with a subroutine $\overline{\sigma}$*. Through analyzing the meta-policy JRS, we show that the performance of each JRS policy can be characterized by the performance of its subroutine, as stated in Theorem 3 below. The proof of Theorem 3 under an irreducibility assumption is given in Section 5, and the proof for the full version is given in Appendix B.3.

Theorem 3 (Conversion Theorem)**.** *Consider a stochastic bin-packing problem in service systems with time-varying job resource requirements. Let the arrival rates be $(\lambda_i r)_{i\in\mathcal{I}}$ and the cost rate budget be $\epsilon > 0$. Let $(\overline{N}, \overline{\sigma}, \pi(\boldsymbol{k}))$ be a solution feasible to the single-server problem $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$. In addition, we assume that the policy $\overline{\sigma}$ is Markovian. Let the infinite-server policy σ be JRS with a subroutine $\overline{\sigma}$. Then under σ, we have*

$$\left|\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}\left[X_{\boldsymbol{k}}\right] - \left\lceil\overline{N}\right\rceil \cdot \mathbb{P}\left(\overline{\boldsymbol{K}} \neq \boldsymbol{0}\right)\right| = O\left(\sqrt{r}\right), \tag{5}$$

$$\left|\sum_{\boldsymbol{k}\neq\boldsymbol{0}} h(\boldsymbol{k})\mathbb{E}\left[X_{\boldsymbol{k}}\right] - \left\lceil\overline{N}\right\rceil \cdot \mathbb{E}\left[h(\overline{\boldsymbol{K}})\right]\right| = O\left(\sqrt{r}\right). \tag{6}$$

As a result,

$$N(\sigma) \leq \left(1 + O\left(r^{-0.5}\right)\right) \cdot \overline{N}, \tag{7}$$

$$C(\sigma) \leq \left(1 + O\left(r^{-0.5}\right)\right) \cdot \epsilon. \tag{8}$$

Optimal Single-Server Policy. Theorem 3 together with the lower bound in Theorem 2 reduces the infinite-server problem $\mathcal{P}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ in (2) to the single-server problem $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ in (3). We can obtain the optimal single-server policy, Single-OPT, by solving a linear program, as stated in the theorem below.

Theorem 4 (Optimality of Single-OPT, Informal). *There exists a linear program $\overline{\mathcal{LP}}((\lambda_i)_{i\in\mathcal{I}}, \epsilon)$ that is equivalent to the single-server problem $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$. In particular, we can construct an optimal Markovian policy for $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ from the optimal solution of $\overline{\mathcal{LP}}((\lambda_i)_{i\in\mathcal{I}}, \epsilon)$.*

Proof of Theorem 4 and details on the construction of the optimal policy are given in Appendix C.

Proof of the Theorem 1 Based on Theorems 2–4. Because Single-OPT (along with an optimal stationary distribution) optimally solves $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$, it achieves the optimal value $\overline{N}^*$. Let σ be JRS with a subroutine Single-OPT, then according to Theorem 3, we have $N(\sigma) \le \left(1 + O\left(r^{-0.5}\right)\right) \cdot \overline{N}^*$ and $C(\sigma) \le \left(1 + O\left(r^{-0.5}\right)\right) \cdot \epsilon$. By Theorem 2, we also have $N^* \ge \overline{N}^*$. So we conclude that JRS with a subroutine Single-OPT is $\left(1 + O(r^{-0.5}), 1 + O(r^{-0.5})\right)$-optimal. □

4 PROPOSED META-POLICY: JOIN-REQUESTING-SERVER (JRS)

In this section, we describe our meta-policy, Join-Requesting-Server (JRS), in full detail. For ease of presentation, we focus on the case where the subroutine policy $\overline{\sigma}$ for JRS is $\boldsymbol{k}^0$*-irreducible*, i.e., under $\overline{\sigma}$, there exists a configuration $\boldsymbol{k}^0$ such that the single-server system can return to $\boldsymbol{k}^0$ from any other configurations (which is equivalent to assuming that the configuration of the single-server system under policy $\overline{\sigma}$ forms a *unichain*). The algorithm for the general case is given in Appendix B.3.

4.1 How the Single-Server Policy Requests Jobs

Before going into the definition of JRS, we first take a closer look at how the Markovian *single-server* policy $\overline{\sigma}$ requests jobs, to avoid potential ambiguity caused by the fact that a single-server policy can request jobs at *any time*. Let a_i denote the number of type i jobs requested, and let $\boldsymbol{a} \triangleq (a_i)_{i\in\mathcal{I}}$. We say $\boldsymbol{a}$ is *feasible* if the total number of jobs on the server does not exceed $K_{\max}$ after adding the jobs. The policy $\overline{\sigma}$ performs one of the following two types of requests based on the current configuration.

- **Reactive requests.** A *reactive request* is triggered by *either an internal transition or a departure*. The changes in the configuration when a reactive request is made can be represented by the diagram
$$\boldsymbol{k} \to \boldsymbol{k}' \to \boldsymbol{k}' + \boldsymbol{a},$$
where $\boldsymbol{k} \to \boldsymbol{k}'$ is due to the internal transition or departure, and $\boldsymbol{k}' \to \boldsymbol{k}' + \boldsymbol{a}$ happens since the policy immediately requests $\boldsymbol{a}$ jobs. The policy $\overline{\sigma}$ specifies a probability distribution over all feasible $\boldsymbol{a}$ when it decides to perform reactive requests for the configuration $\boldsymbol{k}'$.
- **Proactive requests.** A *proactive request* happens on its own, and it happens at a *finite rate* depending on the current configuration of the server. The change of the configuration when a proactive request happens can be represented by the diagram
$$\boldsymbol{k} \to \boldsymbol{k} + \boldsymbol{a}.$$
More specifically, suppose the policy $\overline{\sigma}$ decides to perform proactive requests for a configuration $\boldsymbol{k}$. Then for each feasible $\boldsymbol{a}$, the policy $\overline{\sigma}$ specifies a rate and runs a timer with an exponentially distributed duration with the specified rate. When the timer ticks, $\boldsymbol{a}$ jobs are requested. When the configuration changes, all the timers are canceled and restarted with new rates based on the new configuration.

4.2 Description of Join-Requesting-Server (JRS)

The inputs of JRS include: (i) the single-server policy $\overline{\sigma}$, (ii) the objective value of $\overline{\sigma}$ in the single-server problem (3), denoted as $\overline{N}$, and (iii) the transition rates in the job model.

We first divide the infinite server pool into two sets based on the server index ℓ. Let $L = \lceil \overline{N} \rceil$. We call servers with index $\ell \le L$ *normal servers*; we call servers with index $\ell > L$ *backup servers*. The normal servers are responsible for serving most of the jobs, while the backup servers are activated only to handle overflow jobs (jobs that are not dispatched to normal servers).

The JRS is specified in two steps.

- **Step 1 (Job Requesting on a Normal Server):** We let each normal server request jobs using its subroutine, the single-server policy $\overline{\sigma}$. The input to the policy $\overline{\sigma}$ is what we refer to as the *observed configuration* of the server, which will be further explained below. When $\overline{\sigma}$ requests $\boldsymbol{a} = (a_i)_{i \in \mathcal{I}}$ jobs, a_i type i *tokens* are generated for each $i \in \mathcal{I}$ to store the job requests. The server *pauses* the job requesting process if it already has any type of tokens, and resumes when all the tokens that it generated are removed.
- **Step 2 (Arrival Dispatching):**
  - **Real jobs.** When a type i job arrives, the dispatcher chooses a type i token uniformly at random, removes the token, and assigns the job to the corresponding server. When there are no type i tokens, the dispatcher sends the job to an idle backup server.
  - **Virtual jobs.** When the total number of type i tokens throughout the system exceeds the limit $\eta_{\max} = \lceil \sqrt{L} \rceil$ (called the *token limit*), a type i *virtual arrival* is triggered, which causes the dispatcher to choose a type i token uniformly at random, remove the token, and assign a *virtual job* to the corresponding server. A virtual job has the same transition dynamics as a real job but does not consume physical resources.

The *observed configuration* of a normal server in Step 1 is the configuration resulting from real jobs and virtual jobs combined. That is, it is a vector whose i-th entry represents the total number of real and virtual jobs in phase i on this server. The observed configuration changes when there is a new real or virtual job arrival assigned to the server, or when a real or virtual job on the server has a phase transition or departs. We update the input to the policy $\overline{\sigma}$ when the observed configuration changes. Whenever the observed configuration changes, the policy $\overline{\sigma}$ cancels the exponential timers in progress; but a reactive request from the policy $\overline{\sigma}$ can only be triggered when a real or virtual job on the server has a phase transition or departs.

Intuition behind JRS. To provide a better understanding of the main design ideas of JRS, here we give an intuitive description of how it works. Broadly, servers generate job requests and store unfulfilled requests as tokens; the dispatcher assigns jobs to servers according to the tokens to fulfill job requests. This is the mechanism for matching job arrivals with requests, which is referenced at the end of Section 1.4. However, rather than matching all tokens with job arrivals, JRS opts to convert some of the tokens into virtual jobs to keep the total number of tokens within an upper limit $\eta_{\max}$. By capping the number of tokens, JRS ensures that the job requests generated by each server get fulfilled quickly (either by a real job or a virtual job), and thus the observed configurations of servers maintain proximity to i.i.d. copies of the single-server systems.

The choice of the token limit $\eta_{\max} = \Theta(\sqrt{r})$ balances two key considerations. On the one hand, a smaller $\eta_{\max}$ brings the observed configurations closer to i.i.d. copies of single-server systems. On the other hand, if $\eta_{\max}$ is overly small, the rate of generating virtual jobs becomes high and the probability for a job arrival to see no tokens is also high. As a result, the observed configurations, which include both real and virtual jobs, deviate from the real-job configurations. A more in-depth discussion on the role of tokens and virtual jobs and whether they are fundamental is in Section 5.5.

4.3 Practical considerations in implementing Join-Requesting-Server (JRS)

Computational complexity of JRS. The computational complexity of JRS consists of two components: the *offline* component that computes a single-server policy $\overline{\sigma}$ and its objective value $\overline{N}$, and the *online* component that carries out the two steps of JRS.

The offline component reduces to solving the linear program given in (61) in Appendix C.1, whose number of optimization variables is linear in the number of feasible configurations times the number of phases, i.e., $|\mathcal{K}| \times |\mathcal{I}|$, on a single server. Admittedly, $|\mathcal{K}|$ can be large when a single server has a large quantity of resources and there are many job phases. However, we opt for the view that a single server is not excessively large and the system's scale is primarily captured by the scaling factor r. Therefore, it is advantageous that the computational complexity of this offline component is independent of r.

In the online component, the bulk of the computation is in job requesting and virtual job simulation, which can be executed distributedly on the normal servers. Specifically, each normal server monitors its observed configuration and generates tokens according to the single-server policy $\overline{\sigma}$; additionally, when a virtual job is assigned to the server, the server simulates the dynamics of the virtual job, i.e., generates random variables corresponding to phase transitions and job departure. Backup servers do not need to perform any computation beyond serving jobs.

The scheduler, which stores all the tokens, has two responsibilities in the online component: (i) the scheduler matches each newly arrived job to a token of the same type, chosen uniformly at random, or sends the job to a backup server when there are no tokens of the same type; (ii) the scheduler monitors the number of tokens of each type and assigns virtual jobs when the number of tokens exceeds the limit $\eta_{\max}$.

It is informative to compare the computational complexity of JRS with existing algorithms designed for the traditional setting of stochastic bin-packing in service systems, where the resource requirements are non-time-varying [17, 32–36]. At a high level, these existing algorithms function as follows: upon the arrival of a job, the scheduler checks the current configurations of all servers and assigns the job to a server whose configuration optimizes certain predefined criteria. Among these, the GRAND algorithm [33, 35, 36] stands out for its simplicity and asymptotic optimality. Under GRAND, the scheduler only needs to identify configurations that can accommodate the incoming job and then randomly assigns the job to one of these feasible servers, along with some idle servers. Compared to JRS, GRAND does not have an offline planning component, and individual servers do not perform computation beyond serving jobs. The scheduler's role in GRAND is slightly more complex than in JRS. Consequently, when considering using JRS in settings where job resource requirements are non-time-varying, practitioners should weigh whether the additional computational complexity is warranted.

Model parameter estimation. A limitation of JRS is its dependency on known model parameters, including job arrival rates and phrase transition rates. Such dependency is not present in existing algorithms designed for the setting with non-time-varying resource requirements. In real-world applications, the model parameters can be estimated from workload traces such as [37, 38]. Estimation errors can impact the system's performance, an issue that merits further in-depth investigation in future work. Here, we provide a preliminary result on the performance degradation due to parameter estimation errors. Roughly speaking, suppose that the estimation error in the job arrival rate coefficients λ_i's and the phase transition rates $\mu_{ii'}$'s and $\mu_{i\perp}$'s are bounded by $\delta \geq 0$ (along with an insensitivity assumption on the single-server problem). Then if we use JRS where the single-server policy is obtained by solving for the optimal single-server policy under the estimated parameters, the resulting JRS is $\left(1 + B\delta + O\left(r^{-0.5}\right), 1 + B\delta + O\left(r^{-0.5}\right)\right)$-optimal for any $\delta \leq \delta_{\max}$, where B and

$\delta_{\max}$ are positive constants independent of r. The exact statement is given in Proposition 1 in Appendix D, along with a proof.

Connection to practical algorithms. Recent progress has been made in addressing the issue of low utilization due to time-varying job resource requirements, notably within Google's datacenters, as discussed in [2]. The approach in [2] makes predictions on the future resource requirements of jobs, which lead to a further prediction on the future peak resource requirement on a server if a newly arrived job were to be sent to that server (assuming no future job arrivals). This prediction categorizes each server as either feasible or infeasible for the new job, and this binary outcome is subsequently used by a separate scheduler for job assignment.

Our proposed JRS policy can be viewed as giving more detailed predictions on whether it is suitable for a server to take on new jobs, represented by the tokens. The predictions are optimized by taking into account future job arrivals and the stochastic dynamics of jobs.

5 PROOF OF THEOREM 3 (CONVERSION THEOREM) ASSUMING IRREDUCIBILITY

In this section, we prove Theorem 3 to establish the performance guarantee of JRS. For ease of presentation, we focus on the case where the subroutine policy $\overline{\sigma}$ is $\boldsymbol{k}^0$-irreducible. The proof for the general case is in Appendix B.3.

This section is organized as follows. We first provide some preliminaries in Section 5.1. Then we outline the steps and lemmas needed for the proof in Section 5.2. In Section 5.3, we prove Theorem 3 based on the lemmas. In Section 5.4, we prove one of the lemmas, Lemma 2, where we devise a novel approach to employ Stein's method. Finally, in Section 5.5, we discuss the role of tokens and virtual jobs and their necessity from a proof perspective. The proofs of the rest of the lemmas presented in this section are given in Appendix B.

5.1 Preliminaries

Consider an infinite-server system under the JRS policy. For each normal server ℓ, we describe its status at time t using the following variables: *configuration of real jobs* $\boldsymbol{K}^\ell(t)$ (referred to simply as configuration in previous sections), *tokens* $\boldsymbol{\eta}^\ell(t)$, *configuration of virtual jobs* $\boldsymbol{\zeta}^\ell(t)$, and *observed configuration* $\widehat{\boldsymbol{K}}^\ell(t) \triangleq \boldsymbol{K}^\ell(t) + \boldsymbol{\zeta}^\ell(t)$. We use the superscript "$1:L$" to refer to a certain descriptor of all normal servers, for example, $\widehat{\boldsymbol{K}}^{1:L}(t) \triangleq \left(\widehat{\boldsymbol{K}}^\ell(t)\right)_{\ell=1,2,\dots L}$. The system under JRS is a Markov chain with a unique Markovian representation $((\boldsymbol{K}^\ell(t))_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$. The following lemma shows that the system has a unique stationary distribution (the proof is provided in Appendix B.1).

Lemma 1 (Unique Stationary Distribution)**.** *Consider an infinite-server system under the JRS policy with $\overline{\sigma}$ as its subroutine, where $\overline{\sigma}$ is a single-server policy that is Markovian and $\boldsymbol{k}^0$-irreducible. Then the state of the system $((\boldsymbol{K}^\ell(t))_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$ has a unique stationary distribution.*

Let $\overline{\boldsymbol{K}}^{1:L}(t) \triangleq \left(\overline{\boldsymbol{K}}^\ell(t)\right)_{\ell=1,2,\dots,L}$ be the configuration vector of L i.i.d. copies of the single-server system under $\overline{\sigma}$. As discussed in Section 4.2, we will show that $\boldsymbol{K}^{1:L}(\infty)$ can be approximated by $\overline{\boldsymbol{K}}^{1:L}(\infty)$. In the remainder of this section, we omit the steady-state symbol (∞) for simplicity.

To rigorously discuss the approximation of the steady-state random variables, we define some metrics. Recall that $\mathcal{K} \triangleq \{\boldsymbol{k} \colon \sum_{i\in\mathcal{I}} k_i \le K_{\max}\}$ is the set of feasible single-server configurations. Let $\mathcal{K}^L \triangleq \{\boldsymbol{k}^{1:L} \colon \boldsymbol{k}^\ell \in \mathcal{K}, \forall \ell\}$ be the set of feasible configurations for all normal servers. We use $\|\cdot\|$ to denote the L^1 norm in both space $\mathcal{K}$ and space $\mathcal{K}^L$:

$$\|\boldsymbol{k} - \boldsymbol{k}'\| = \textstyle\sum_{i\in\mathcal{I}} \left|k_i - k_i'\right|, \quad \text{for } \boldsymbol{k}, \boldsymbol{k}' \in \mathcal{K},$$
$$\|\boldsymbol{k}^{1:L} - \boldsymbol{k}'^{1:L}\| = \textstyle\sum_{\ell=1}^{L} \|\boldsymbol{k}^\ell - \boldsymbol{k}'^\ell\|, \quad \text{for } \boldsymbol{k}^{1:L}, \boldsymbol{k}'^{1:L} \in \mathcal{K}^L.$$

For any two random variables $\boldsymbol{U}^a, \boldsymbol{U}^b \in \mathcal{K}^L$, their closeness will be measured in terms of Wasserstein distance as follows:

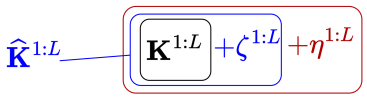
$$d(\boldsymbol{U}^a, \boldsymbol{U}^b) \triangleq \sup_{f \in \mathrm{Lip}(1)} \left\{ \mathbb{E}\left[f\left(\boldsymbol{U}^a\right)\right] - \mathbb{E}[f(\boldsymbol{U}^b)] \right\},$$

where the supremum is taken over the all Lipschitz-1 functions from $\mathcal{K}^L$ to $\mathbb{R}$.

5.2 Steps and Lemmas Needed for the Proof of Theorem 3 Assuming Irreducibility

Our goal is to show that the steady-state distribution of the normal servers' real-job configurations $\boldsymbol{K}^{1:L}$ is close to the steady-state distribution of i.i.d. copies of the single-server systems $\overline{\boldsymbol{K}}^{1:L}$ in Wasserstein distance, and that the backup servers are almost empty as the arrival rate gets large. More formally, we aim to show that $d(\boldsymbol{K}^{1:L}, \overline{\boldsymbol{K}}^{1:L}) = O(\sqrt{r})$ and $\sum_{\ell=L+1}^{\infty} \sum_{i \in \mathcal{I}} K_i^\ell = O(\sqrt{r})$ as $r \to \infty$. These two bounds provide the performance guarantee claimed in Theorem 3.

Instead of directly looking into the distribution of real-job configuration $\boldsymbol{K}^{1:L}$, we show that the distribution of each of the three sums, $\boldsymbol{K}^{1:L} + \boldsymbol{\zeta}^{1:L} + \boldsymbol{\eta}^{1:L}$, $\boldsymbol{K}^{1:L} + \boldsymbol{\zeta}^{1:L}$, and $\boldsymbol{K}^{1:L}$, can be approximated by the distribution of $\overline{\boldsymbol{K}}^{1:L}$ in Wasserstein distance. The approximation result for each sum helps us derive the approximation result for the sum with one fewer term. The result that the backup servers are almost empty also follows from these approximations. This sequence of approximations is illustrated in the figure below, where recall that $\widehat{\boldsymbol{K}}^\ell(t) \triangleq \boldsymbol{K}^\ell(t) + \boldsymbol{\zeta}^\ell(t)$ is the observed configuration.



A crucial observation that leads to this stepwise proof is that the process $(\widehat{\boldsymbol{K}}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$ forms a Markov chain on its own. This is because real jobs and virtual jobs have the same transition dynamics and are indistinguishable by the subroutine when requesting jobs. Moreover, by the construction of JRS, the Markov chain $(\widehat{\boldsymbol{K}}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$ governs the dynamics of the virtual-job configurations $\boldsymbol{\zeta}^{1:L}(t)$ and the configurations on backup servers.

Our proof consists of two steps. In **Step 1**, we focus on the process $(\widehat{\boldsymbol{K}}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$. We show that $d(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}, \overline{\boldsymbol{K}}^{1:L}) = O(\sqrt{r})$, which immediately implies $d(\widehat{\boldsymbol{K}}^{1:L}, \overline{\boldsymbol{K}}^{1:L}) = O(\sqrt{r})$ because we have limited the total number of tokens to $O(\sqrt{r})$. In **Step 2**, we use the approximation result for $\widehat{\boldsymbol{K}}^{1:L}$ in **Step 1** to show that the total number of virtual jobs, $\sum_{i \in \mathcal{I}} \sum_{\ell=1}^{L} \zeta_i^\ell$, and the total number of jobs on backup servers are both $O(\sqrt{r})$. Recall that $\boldsymbol{K}^{1:L} = \widehat{\boldsymbol{K}}^{1:L} - \boldsymbol{\zeta}^{1:L}$, so we get $d(\boldsymbol{K}^{1:L}, \overline{\boldsymbol{K}}^{1:L}) = O(\sqrt{r})$.

Next, we state the specific lemmas.

Step 1. Lemma 2 below bounds the Wasserstein distance between $\widehat{\boldsymbol{K}}^{1:L}$ and $\overline{\boldsymbol{K}}^{1:L}$.

Lemma 2. *Under the conditions of Theorem 3 and $\overline{\sigma}$ being $\boldsymbol{k}^0$-irreducible, we have*

$$d\left(\widehat{\boldsymbol{K}}^{1:L}, \overline{\boldsymbol{K}}^{1:L}\right) = O\left(\sqrt{r}\right).$$

The key challenge for proving Lemma 2 is that the job dispatching decisions are based on the configurations of *all normal servers*, which creates dependencies among the transitions of different servers. The key idea that helps us overcome this challenge is to consider the sum $\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}$, which remains unchanged under job arrivals regardless of dispatching decisions. Observe that $\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}$ has *decoupled* dynamics across servers because it is only changed by internal phase transitions, departures, and requests of new jobs, which happen independently on each server. This helps us prove $d(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}, \overline{\boldsymbol{K}}^{1:L}) = O\left(\sqrt{r}\right)$, which implies Lemma 2, as argued earlier in the section.

Formally, the proof of Lemma 2 makes use of Stein's method (see, e.g., [3–5]) to compare $\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}$ with $\overline{\boldsymbol{K}}^{1:L}$. Stein's method usually consists of three steps: generator comparison, Stein factor bound,

and moment bound. In our case, due to the finiteness of the state space $\mathcal{K}$, we only need to do the generator comparison and the Stein factor bound. In the generator comparison step, we show that the instantaneous transition rates of $\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}$ match with those of $\overline{\boldsymbol{K}}^{1:L}$; in the Stein factor bound step, we show that small difference in the transition rates of $\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}$ and $\overline{\boldsymbol{K}}^{1:L}$ does not cause much increase in the overall distance of the distributions. The detailed proof is in Section 5.4.

Step 2. We establish Lemma 3 and Lemma 4 below, which bound the steady-state expected number of virtual jobs and the jobs on backup servers.

Lemma 3. *Under the conditions of Theorem 3 and $\overline{\sigma}$ being $\boldsymbol{k}^0$-irreducible, for each $i \in \mathcal{I}$, the steady-state expected number of virtual jobs of type i is of the order $O(\sqrt{r})$, i.e.,*

$$\mathbb{E}\left[\sum_{\ell=1}^{L} \zeta_i^\ell\right] = O\left(\sqrt{r}\right).$$

Lemma 4. *Under the conditions of Theorem 3 and $\overline{\sigma}$ being $\boldsymbol{k}^0$-irreducible, for each $i \in \mathcal{I}$, the steady-state expected number of type i jobs on backup servers is of the order $O(\sqrt{r})$, i.e.,*

$$\mathbb{E}\left[\sum_{\ell=L+1}^{\infty} K_i^\ell\right] = O\left(\sqrt{r}\right).$$

The key idea for proving Lemma 3 and Lemma 4 is that by the characterization of $\widehat{\boldsymbol{K}}^{1:L}$ in Lemma 2 and the fact that the job requests are made based on $\widehat{\boldsymbol{K}}^{1:L}$, we can show that the rate of requesting jobs is approximately equal to the arrival rate for each job type. Therefore, the number of tokens rarely reaches 0 or $\eta_{\max}$. This implies the rarity of virtual jobs and jobs on backup servers. The proofs are provided in Appendix B.2.

5.3 Proof of Theorem 3 Assuming Irreducibility Based on Lemmas 1–4.

Proof. First we show that Lemmas 2 and 3 imply the closeness between $\boldsymbol{K}^{1:L}$ and $\overline{\boldsymbol{K}}^{1:L}$. By Lemma 2, for any $f \in \mathrm{Lip}(1)$, we have $\mathbb{E}\big[f(\overline{\boldsymbol{K}}^{1:L})\big] - \mathbb{E}\big[f(\widehat{\boldsymbol{K}}^{1:L})\big] = O\left(\sqrt{r}\right)$. By Lemma 3, $\mathbb{E}\left[\sum_{\ell=1}^{L} \zeta_i^\ell\right] = O\left(\sqrt{r}\right)$. Recall that $\widehat{\boldsymbol{K}}^\ell = \boldsymbol{K}^\ell + \boldsymbol{\zeta}^\ell$, so $\mathbb{E}\big[f(\widehat{\boldsymbol{K}}^{1:L})\big] - \mathbb{E}\big[f(\boldsymbol{K}^{1:L})\big] = O\left(\sqrt{r}\right)$. Therefore,

$$\mathbb{E}\left[f(\overline{\boldsymbol{K}}^{1:L})\right] - \mathbb{E}\left[f(\boldsymbol{K}^{1:L})\right] = O\left(\sqrt{r}\right). \tag{9}$$

Now we prove (5), the bound on the expected number of the active servers, by taking a suitable f in (9). Observe that

$$\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}\Big[X_{\boldsymbol{k}}\Big] - L\cdot\mathbb{P}(\overline{\boldsymbol{K}}\neq 0) = \mathbb{E}\Big[\sum_{\ell=1}^{L} \mathbb{1}_{\{\boldsymbol{K}^\ell\neq\boldsymbol{0}\}}\Big] - \mathbb{E}\Big[\sum_{\ell=1}^{L} \mathbb{1}_{\{\overline{\boldsymbol{K}}^\ell\neq\boldsymbol{0}\}}\Big] + \mathbb{E}\Big[\sum_{\ell=L+1}^{\infty} \mathbb{1}_{\{\boldsymbol{K}^\ell\neq\boldsymbol{0}\}}\Big], \tag{10}$$

where the last term on RHS is $O\left(\sqrt{r}\right)$ by Lemma 4. To show that the difference between the first two terms on the RHS of (10) are also $O\left(\sqrt{r}\right)$, consider $f_1(\boldsymbol{k}^{1:L}) \triangleq \sum_{\ell=1}^{L} \mathbb{1}_{\{\boldsymbol{k}^\ell\neq\boldsymbol{0}\}}$. Because

$$\left|f_1(\boldsymbol{k}^{1:L}) - f_1(\boldsymbol{k}'^{1:L})\right| = \left|\sum_{\ell=1}^{L}\left(\mathbb{1}_{\{\boldsymbol{k}^\ell\neq\boldsymbol{0}\}} - \mathbb{1}_{\{\boldsymbol{k}'^\ell\neq\boldsymbol{0}\}}\right)\right| \le \sum_{\ell=1}^{L} \mathbb{1}_{\{\boldsymbol{k}^\ell\neq\boldsymbol{k}'^\ell\}} \le \|\boldsymbol{k}^{1:L} - \boldsymbol{k}'^{1:L}\|,$$

for any $\boldsymbol{k}^{1:L}, \boldsymbol{k}'^{1:L} \in \mathcal{K}^L$, we have $f_1 \in \mathrm{Lip}(1)$. By (9), $\mathbb{E}\big[f_1(\boldsymbol{K}^\ell)\big] - \mathbb{E}\big[\sum_{\ell=1}^{L} f_1(\overline{\boldsymbol{K}}^\ell)\big] = O\left(\sqrt{r}\right)$. Therefore, $\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}\big[X_{\boldsymbol{k}}\big] - L\cdot\mathbb{P}\big(\overline{\boldsymbol{K}}\neq 0\big) = O\left(\sqrt{r}\right)$. Recall that $L = \left\lceil \overline{N} \right\rceil$, so we get (5).

Similarly, to prove (6), we observe that

$$\sum_{\boldsymbol{k}\neq\boldsymbol{0}} h(\boldsymbol{k})\mathbb{E}[X_{\boldsymbol{k}}] - L\cdot\mathbb{E}\Big[h(\overline{\boldsymbol{K}})\Big] = \mathbb{E}\Big[\sum_{\ell=1}^{L} h(\boldsymbol{K}^\ell)\Big] - \mathbb{E}\Big[\sum_{\ell=1}^{L} h(\overline{\boldsymbol{K}}^\ell)\Big] + \mathbb{E}\Big[\sum_{\ell=L+1}^{\infty} h(\boldsymbol{K}^\ell)\Big]. \tag{11}$$

The last term of (11) can be bounded as $\mathbb{E}\Big[\sum_{\ell=L+1}^{\infty} h(\boldsymbol{K}^\ell)\Big] \le \mathbb{E}\Big[\sum_{\ell=L+1}^{\infty} \mathbb{1}_{\{K_i^\ell \neq \mathbf{0}\}}\Big] \cdot \max_{\boldsymbol{k}\in\mathcal{K}} h(\boldsymbol{k})$, which is $O\left(\sqrt{r}\right)$ by Lemma 4 and the fact that $\mathcal{K}$ is a finite set. To show that the difference between the first two terms on the RHS of (11) is also $O\left(\sqrt{r}\right)$, consider $f_2(\boldsymbol{k}^{1:L}) = \frac{1}{\Gamma}\sum_{\ell=1}^{L} h(\boldsymbol{k}^\ell)$, where Γ is the Lipschitz constant of $h(\cdot)$. Because

$$\left|f_2(\boldsymbol{k}^{1:L}) - f_2(\boldsymbol{k}'^{1:L})\right| = \frac{1}{\Gamma}\left|\sum_{\ell=1}^{L}(h(\boldsymbol{k}^\ell) - h(\boldsymbol{k}'^\ell))\right| \le \sum_{\ell=1}^{L}\|\boldsymbol{k}^\ell - \boldsymbol{k}'^\ell\| = \|\boldsymbol{k}^{1:L} - \boldsymbol{k}'^{1:L}\|,$$

for any $\boldsymbol{k}^{1:L}, \boldsymbol{k}'^{1:L} \in \mathcal{K}^L$, we have $f_2 \in \mathrm{Lip}(1)$. By (9), $\mathbb{E}\big[f_2(\boldsymbol{K}^\ell)\big] - \mathbb{E}\Big[\sum_{\ell=1}^{L} f_2(\overline{\boldsymbol{K}}^\ell)\Big] = O\left(\sqrt{r}\right)$. Therefore, $\sum_{\boldsymbol{k}\neq\mathbf{0}} h(\boldsymbol{k})\mathbb{E}[X_{\boldsymbol{k}}] - L\cdot\mathbb{E}\left[h(\overline{\boldsymbol{K}})\right] = O\left(\sqrt{r}\right)$. Recall that $L = \left\lceil \overline{N} \right\rceil$, so we get (6).

To show (7) and (8), noting that $\left\lceil \overline{N} \right\rceil = \Theta(r)$, we have

$$N(\sigma) = \textstyle\sum_{\boldsymbol{k}\neq\mathbf{0}} \mathbb{E}[X_{\boldsymbol{k}}] \le \left\lceil \overline{N} \right\rceil + O\left(\sqrt{r}\right) = \left(1 + O\left(r^{-0.5}\right)\right)\cdot \overline{N},$$

$$C(\sigma) = \frac{\sum_{\boldsymbol{k}\neq\mathbf{0}} h(\boldsymbol{k})\mathbb{E}[X_{\boldsymbol{k}}]}{\sum_{\boldsymbol{k}\neq\mathbf{0}} \mathbb{E}[X_{\boldsymbol{k}}]} = \frac{\sum_{\boldsymbol{k}\neq\mathbf{0}} h(\boldsymbol{k})\pi(\boldsymbol{k}) + O\left(r^{-0.5}\right)}{1 - \pi(\mathbf{0}) + O\left(r^{-0.5}\right)} \le \left(1 + O\left(r^{-0.5}\right)\right)\cdot\epsilon,$$

where in the last inequality we have used the fact that $\epsilon > 0$. This completes the proof.

□

5.4 More Details on the System and Proof of Lemma 2

To bound the distance between $\overline{\boldsymbol{K}}^{1:L}$ and $\widehat{\boldsymbol{K}}^{1:L}$, observe that because $f \in \mathrm{Lip}(1)$ and $\sum_{\ell=1}^{L}\sum_{i\in\mathcal{I}} \eta_i^\ell = O\left(\sqrt{r}\right)$, it suffices to bound bound the Wasserstein distance between $\overline{\boldsymbol{K}}^{1:L}$ and $\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}$, i.e.,

$$\sup_{f\in\mathrm{Lip}(1)} \left\{\mathbb{E}\left[f\left(\overline{\boldsymbol{K}}^{1:L}\right)\right] - \mathbb{E}\left[f\left(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}\right)\right]\right\} = O\left(\sqrt{r}\right), \tag{12}$$

where $f\left(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}\right)$ is a valid expression because $\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L} \in \mathcal{K}$ as discussed in Remark 1 below.

5.4.1 ***More Details on System Dynamics and Generator.*** To prepare for the proof, we first look into the dynamics of the two systems under study. In particular, we write out the *generators* of $\overline{\boldsymbol{K}}^{1:L}(t)$ and $(\widehat{\boldsymbol{K}}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$, which are used in the Stein's method arguments.

We first examine the dynamics of the single-server system under Markovian policy $\overline{\sigma}$. Four types of events change a single-server system's configuration: internal transitions, departures, reactive requests, and proactive requests (see Section 4.1). The change of configuration due to any event can be represented by the diagram

$$\boldsymbol{k} \to \boldsymbol{k}' \to \boldsymbol{k}' + \boldsymbol{a},$$

where the arrow $\boldsymbol{k} \to \boldsymbol{k}'$ denotes an internal transition or a departure from configuration $\boldsymbol{k}$ to $\boldsymbol{k}'$ if $\boldsymbol{k} \neq \boldsymbol{k}'$; the arrow $\boldsymbol{k}' \to \boldsymbol{k}' + \boldsymbol{a}$ denotes a reactive request that adds $\boldsymbol{a}$ jobs to the system if $\boldsymbol{k} \neq \boldsymbol{k}'$, and denotes a proactive request if $\boldsymbol{k} = \boldsymbol{k}'$. We call the above change of configuration a *transition*, and denote its rate as $\gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$. Let $E(\boldsymbol{k})$ denote the set of possible $(\boldsymbol{k}', \boldsymbol{a})$ pairs in a transition.

We define the total transition rate at configuration $\boldsymbol{k}$ as $\gamma_{\boldsymbol{k}} \triangleq \sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\boldsymbol{k})} \gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$, and define the maximal transition rate $\gamma_{\max} = \max_{\boldsymbol{k}\in\mathcal{K}} \gamma_{\boldsymbol{k}}$. Since $\mathcal{K}$ is a finite set, we have $\gamma_{\max} < \infty$. Also, observe that the request rate of type i jobs is given by

$$\overline{\lambda}_i \triangleq \sum_{\boldsymbol{k}} \sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\boldsymbol{k})} \gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}\, a_i \cdot \pi(\boldsymbol{k}), \tag{13}$$

where π denotes the stationary distribution of single-server configuration under policy $\overline{\sigma}$.

Next, we focus on the dynamics of L i.i.d. copies of single-server systems. Consider the generator $\overline{G}$ of the corresponding Markov chain $\{\overline{\boldsymbol{K}}^{1:L}(t)\}$, which is a linear operator on functions $g\colon \mathcal{K}^L \to \mathbb{R}$ defined as:

$$\overline{G}g(\boldsymbol{k}^{1:L}) \triangleq \frac{d}{dt}\mathbb{E}\left[g\left(\overline{\boldsymbol{K}}^{1:L}(t)\right)\middle|\overline{\boldsymbol{K}}^{1:L}(0)=\boldsymbol{k}^{1:L}\right]\Big|_{t=0}, \tag{14}$$

and we call the resulting function $\overline{G}g(\cdot)$ the *drift* of $g(\cdot)$. Based on the transition rates defined above, we have

$$\overline{G}g(\boldsymbol{k}^{1:L}) = \sum_{\ell=1}^{L}\sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\boldsymbol{k}^\ell)} \gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})}\left(g(\cdot,\boldsymbol{k}'+\boldsymbol{a},\cdot)-g(\cdot,\boldsymbol{k}^\ell,\cdot)\right), \tag{15}$$

where $g(\cdot,\boldsymbol{k}'+\boldsymbol{a},\cdot)-g(\cdot,\boldsymbol{k}^\ell,\cdot)$ is a shorthand for $g(\boldsymbol{k}^1,\dots,\boldsymbol{k}^{\ell-1},\boldsymbol{k}'+\boldsymbol{a},\boldsymbol{k}^{\ell+1},\dots,\boldsymbol{k}^L)-g(\boldsymbol{k}^{1:L})$, i.e., we use $\cdot$ to omit the entries that agree with $\boldsymbol{k}^{1:L}$.

Similarly, for the infinite-server system, consider the generator $\widehat{G}$ of $(\widehat{\boldsymbol{K}}^{1:L}(t),\boldsymbol{\eta}^{1:L}(t))$ defined as

$$\widehat{G}\psi(\boldsymbol{k}^{1:L},\boldsymbol{\eta}^{1:L}) \triangleq \frac{d}{dt}\mathbb{E}\left[\psi\left(\widehat{\boldsymbol{K}}^{1:L}(t),\boldsymbol{\eta}^{1:L}(t)\right)\middle|\widehat{\boldsymbol{K}}^{1:L}(0)=\boldsymbol{k}^{1:L},\boldsymbol{K}^{1:L}(0)=\boldsymbol{\eta}^{1:L}\right]\Big|_{t=0}, \tag{16}$$

for any function $\psi\colon (\mathcal{K}\times\mathcal{K})^L \to \mathbb{R}$. The drift of ψ under $\widehat{G}$ turns out to have a similar decoupled form as $\overline{G}g$: observe that for each ℓ, the transition of $(\widehat{\boldsymbol{K}}^\ell(t),\boldsymbol{\eta}^\ell(t))$ from $(\boldsymbol{k},\boldsymbol{\eta})$ to $(\boldsymbol{k}',\boldsymbol{\eta}+\boldsymbol{a}\mathbb{1}_{\{\boldsymbol{\eta}=\boldsymbol{0}\}})$ occurs at the rate $\gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ for each $(\boldsymbol{k}',\boldsymbol{a})\in E(\boldsymbol{k})$, and any real or virtual job arrivals do not change the sum $\widehat{\boldsymbol{K}}^\ell(t)+\boldsymbol{\eta}^\ell(t)$. Consider any function $g\colon \mathcal{K}^L\to\mathbb{R}$ and the function $\psi(\boldsymbol{k}^{1:L},\boldsymbol{\eta}^{1:L}) = g(\boldsymbol{k}^{1:L}+\boldsymbol{\eta}^{1:L})$.

$$\begin{aligned}\widehat{G}\psi(\boldsymbol{k}^{1:L},\boldsymbol{\eta}^{1:L}) = &\sum_{\ell=1}^{L}\sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\boldsymbol{k}^\ell)} \gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})}\left(g(\cdot,\boldsymbol{k}'+\boldsymbol{a},\cdot)-g(\cdot,\boldsymbol{k}^\ell,\cdot)\right)\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\\ &+\sum_{\ell=1}^{L}\sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\boldsymbol{k}^\ell)} \gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})}\left(g(\cdot,\boldsymbol{k}'+\boldsymbol{\eta}^\ell,\cdot)-g(\cdot,\boldsymbol{k}^\ell+\boldsymbol{\eta}^\ell,\cdot)\right)\mathbb{1}_{\{\boldsymbol{\eta}^\ell\neq\boldsymbol{0}\}}.\end{aligned} \tag{17}$$

In this context $g(\cdot,\boldsymbol{k}'+\boldsymbol{\eta}^\ell,\cdot)-g(\cdot,\boldsymbol{k}^\ell+\boldsymbol{\eta}^\ell,\cdot)$ is a shorthand for $g(\boldsymbol{k}^1+\boldsymbol{\eta}^1,\dots,\boldsymbol{k}^{\ell-1}+\boldsymbol{\eta}^{\ell-1},\boldsymbol{k}'+\boldsymbol{\eta}^\ell,\boldsymbol{k}^{\ell+1}+\boldsymbol{\eta}^{\ell+1},\dots,\boldsymbol{k}^L+\boldsymbol{\eta}^L)-g(\boldsymbol{k}^{1:L}+\boldsymbol{\eta}^{1:L})$. In other words, we use $\cdot$ to omit the entries of g's input that agree with the corresponding entries of $\boldsymbol{k}^{1:L}+\boldsymbol{\eta}^{1:L}$.

Remark 1. In (17), although g is only defined on the domain $\mathcal{K}^L$, it is valid to write $\boldsymbol{k}^{1:L}+\boldsymbol{\eta}^{1:L}$ as its input because we always have $\widehat{\boldsymbol{K}}^\ell+\boldsymbol{\eta}^\ell\in\mathcal{K}$, i.e., the total number of real jobs, virtual jobs, and tokens on a normal server never exceeds $K_{\max}$. To see why this is true, the single-server policy $\overline{\sigma}$ requests jobs only when there are no tokens on the server, and it will not request more than $K_{\max}-n$ jobs if there are already n real and virtual jobs on the server.

5.4.2 *Proof of Lemma 2.*

Proof. **Generator Comparison.** For any $f\in\mathrm{Lip}(1)$, consider the Poisson equation (see, e.g., [3]) that solves for $g_f\colon\mathcal{K}^L\to\mathbb{R}$:

$$\mathbb{E}\left[f\left(\overline{\boldsymbol{K}}^{1:L}\right)\right]-f\left(\boldsymbol{k}^{1:L}\right)=\overline{G}g_f\left(\boldsymbol{k}^{1:L}\right). \tag{18}$$

We let $\boldsymbol{k}^{1:L}=\widehat{\boldsymbol{K}}^{1:L}+\boldsymbol{\eta}^{1:L}$ in (18) and take the expectation. This results in

$$\mathbb{E}\left[f\left(\overline{\boldsymbol{K}}^{1:L}\right)\right]-\mathbb{E}\left[f\left(\widehat{\boldsymbol{K}}^{1:L}+\boldsymbol{\eta}^{1:L}\right)\right]=\mathbb{E}\left[\overline{G}g_f\left(\widehat{\boldsymbol{K}}^{1:L}+\boldsymbol{\eta}^{1:L}\right)\right]. \tag{19}$$

On the other hand, because $(\widehat{\boldsymbol{K}}^{1:L}(t),\boldsymbol{\eta}^{1:L}(t))$ is a finite-state Markov chain, we have

$$\mathbb{E}\left[\widehat{G}\psi_f\left(\widehat{\boldsymbol{K}}^{1:L},\boldsymbol{\eta}^{1:L}\right)\right]=0, \tag{20}$$

where ψ_f is given by $\psi_f\big(\widehat{\boldsymbol{K}}^{1:L}, \boldsymbol{\eta}^{1:L}\big) = g_f(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L})$. Subtracting (20) from (19), we get

$$\mathbb{E}\left[f\big(\overline{\boldsymbol{K}}^{1:L}\big)\right] - \mathbb{E}\left[f\big(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}\big)\right] = \mathbb{E}\left[\overline{G} g_f\big(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}\big) - \widehat{G}\psi_f\big(\widehat{\boldsymbol{K}}^{1:L}, \boldsymbol{\eta}^{1:L}\big)\right]. \tag{21}$$

We want to show that $\overline{G}$ and $\widehat{G}$ are close so that we can bound the RHS of (21).

Now we plug the formula of the generators in (15) and (17) into the RHS of (21) and get

$$\begin{aligned}
&\left|\overline{G} g_f\big(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}\big) - \widehat{G}\psi_f\big(\widehat{\boldsymbol{K}}^{1:L}, \boldsymbol{\eta}^{1:L}\big)\right| \\
&\overset{\text{(i)}}{=} \left| \sum_{\ell=1}^{L} \sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\widehat{\boldsymbol{K}}^\ell+\boldsymbol{\eta}^\ell)} \gamma_{\widehat{\boldsymbol{K}}^\ell+\boldsymbol{\eta}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(g_f\big(\cdot, \boldsymbol{k}'+\boldsymbol{a}, \cdot\big) - g_f\big(\cdot, \widehat{\boldsymbol{K}}^\ell + \boldsymbol{\eta}^\ell, \cdot\big)\right) \cdot \mathbb{1}_{\{\boldsymbol{\eta}^\ell \neq \boldsymbol{0}\}} \right. \\
&\quad \left. - \sum_{\ell=1}^{L} \sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\widehat{\boldsymbol{K}}^\ell)} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(g_f\big(\cdot, \boldsymbol{k}'+\boldsymbol{\eta}^\ell, \cdot\big) - g_f\big(\cdot, \widehat{\boldsymbol{K}}^\ell + \boldsymbol{\eta}^\ell, \cdot\big)\right) \cdot \mathbb{1}_{\{\boldsymbol{\eta}^\ell \neq \boldsymbol{0}\}} \right| \\
&\overset{\text{(ii)}}{\leq} \sum_{\ell=1}^{L} \gamma_{\widehat{\boldsymbol{K}}^\ell+\boldsymbol{\eta}^\ell} \cdot \sup_{(\boldsymbol{k}',\boldsymbol{a})\in E(\widehat{\boldsymbol{K}}^\ell+\boldsymbol{\eta}^\ell)} \left|g_f\big(\cdot, \boldsymbol{k}'+\boldsymbol{a}, \cdot\big) - g_f\big(\cdot, \widehat{\boldsymbol{K}}^\ell + \boldsymbol{\eta}^\ell, \cdot\big)\right| \cdot \mathbb{1}_{\{\boldsymbol{\eta}^\ell \neq \boldsymbol{0}\}} \\
&\quad + \sum_{\ell=1}^{L} \gamma_{\widehat{\boldsymbol{K}}^\ell} \cdot \sup_{(\boldsymbol{k}',\boldsymbol{a})\in E(\widehat{\boldsymbol{K}}^\ell)} \left|g_f\big(\cdot, \boldsymbol{k}'+\boldsymbol{\eta}^\ell, \cdot\big) - g_f\big(\cdot, \widehat{\boldsymbol{K}}^\ell + \boldsymbol{\eta}^\ell, \cdot\big)\right| \cdot \mathbb{1}_{\{\boldsymbol{\eta}^\ell \neq \boldsymbol{0}\}} \\
&\overset{\text{(iii)}}{\leq} 2\gamma_{\max} \cdot \sum_{\ell=1}^{L} \sup_{\boldsymbol{k}'\in\mathcal{K}} \left|g_f\big(\cdot, \boldsymbol{k}', \cdot\big) - g_f\big(\cdot, \widehat{\boldsymbol{K}}^\ell + \boldsymbol{\eta}^\ell, \cdot\big)\right| \cdot \mathbb{1}_{\{\boldsymbol{\eta}^\ell \neq \boldsymbol{0}\}} \\
&\leq 2\gamma_{\max} \cdot \sup_{\boldsymbol{k},\boldsymbol{k}'\in\mathcal{K}} \left|g_f\big(\cdot, \boldsymbol{k}', \cdot\big) - g_f\big(\cdot, \boldsymbol{k}, \cdot\big)\right| \cdot \sum_{\ell=1}^{L} \mathbb{1}_{\{\boldsymbol{\eta}^\ell \neq \boldsymbol{0}\}},
\end{aligned} \tag{22}$$

where in $g_f\big(\cdot, \boldsymbol{k}', \cdot\big) - g_f\big(\cdot, \boldsymbol{k}, \cdot\big)$ we have omitted the entries that agree with $\boldsymbol{k}^{1:L}$. The equality (i) is true because each of the ℓ-th terms in $\overline{G}$ and $\widehat{G}$ is equal if $\boldsymbol{\eta}^\ell = \boldsymbol{0}$. For the inequalities (ii) and (iii), recall that $\gamma_{\boldsymbol{k}}$ is the total transition rate given by $\gamma_{\boldsymbol{k}} \triangleq \sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\boldsymbol{k})} \gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$, and $\gamma_{\max} = \max_{\boldsymbol{k}\in\mathcal{K}} \gamma_{\boldsymbol{k}}$. Observe that

$$\sum_{\ell=1}^{L} \mathbb{1}_{\{\boldsymbol{\eta}^\ell \neq \boldsymbol{0}\}} \leq \sum_{\ell=1}^{L} \sum_{i\in\mathcal{I}} \eta_i^\ell \leq |\mathcal{I}| \cdot \eta_{\max} = O\left(\sqrt{r}\right).$$

Therefore (22) can be further bounded by

$$\begin{aligned}
\left|\overline{G} g_f\big(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}\big) - \widehat{G}\psi_f\big(\widehat{\boldsymbol{K}}^{1:L}, \boldsymbol{\eta}^{1:L}\big)\right| &\leq 2\gamma_{\max} \cdot \sup_{\boldsymbol{k},\boldsymbol{k}'\in\mathcal{K}} \left|g_f\big(\cdot, \boldsymbol{k}', \cdot\big) - g_f\big(\cdot, \boldsymbol{k}, \cdot\big)\right| \cdot \sum_{\ell=1}^{L} \sum_{i\in\mathcal{I}} \eta_i^\ell \\
&\leq 2\gamma_{\max} \cdot \sup_{\boldsymbol{k},\boldsymbol{k}'\in\mathcal{K}} \left|g_f\big(\cdot, \boldsymbol{k}', \cdot\big) - g_f\big(\cdot, \boldsymbol{k}, \cdot\big)\right| \cdot O\left(\sqrt{r}\right).
\end{aligned}$$

To prove (12), it remains to show that

$$\sup_{\boldsymbol{k},\boldsymbol{k}'\in\mathcal{K}} \left|g_f(\cdot, \boldsymbol{k}', \cdot) - g_f(\cdot, \boldsymbol{k}, \cdot)\right| = O(1). \tag{23}$$

Stein Factor Bound. To prove (23), observe that the following $g_f(\cdot)$ is a solution to the Poisson equation (18):

$$g_f\big(\boldsymbol{k}^{1:L}\big) = \int_0^\infty \mathbb{E}\left[\left(f\big(\overline{\boldsymbol{K}}^{1:L}(t)\big) - \mathbb{E}\left[f\big(\overline{\boldsymbol{K}}^{1:L}\big)\right]\right) \middle| \overline{\boldsymbol{K}}^{1:L}(0) = \boldsymbol{k}^{1:L}\right] dt. \tag{24}$$

This allows us to bound the difference of g_f using coupling. Specifically, we define the coupling of two systems, each consisting of L i.i.d. copies of the single-server system under $\overline{\sigma}$. The two systems are initialized with configurations $(\cdot, \boldsymbol{k}', \cdot)$ and $(\cdot, \boldsymbol{k}, \cdot)$ that only differ at the ℓ-th server, where we omit the entries that agree with $\boldsymbol{k}^{1:L}$. Let $\left(\overline{\boldsymbol{K}}^{1:L,1}(t), \overline{\boldsymbol{K}}^{1:L,2}(t)\right)$ be the joint configuration of the two systems, which is actually $2L$ i.i.d. copies of the single-server system. As a result, we can specify the couplings $(\overline{\boldsymbol{K}}^{\ell',1}(t), \overline{\boldsymbol{K}}^{\ell',2}(t))$ for different ℓ' separately. For $\ell' \neq \ell$, the corresponding server in the two systems have the same initial configurations, so we can always keep their configurations identical. For the ℓ-th servers, we let them evolve independently following their own dynamics until a stopping time τ_{mix} when their configurations become the same. After that, we can use coupling to keep their configurations identical. Under this coupling, it is not hard to see that

$$\begin{aligned}
\left|g_f(\cdot, \boldsymbol{k}', \cdot) - g_f(\cdot, \boldsymbol{k}, \cdot)\right| &= \left|\int_0^\infty \mathbb{E}\left[f\left(\overline{\boldsymbol{K}}^{1:L,1}(t)\right) - f\left(\overline{\boldsymbol{K}}^{1:L,2}(t)\right)\right] dt\right| \\
&\leq \mathbb{E}\left[\int_0^\infty \left|f\left(\overline{\boldsymbol{K}}^{1:L,1}(t)\right) - f\left(\overline{\boldsymbol{K}}^{1:L,2}(t)\right)\right| dt\right] \\
&\leq \mathbb{E}\left[\int_0^\infty \sum_{\ell'=1}^{L} \left\|\overline{\boldsymbol{K}}^{\ell',1}(t) - \overline{\boldsymbol{K}}^{\ell',2}(t)\right\| dt\right] \\
&= \mathbb{E}\left[\int_0^{\tau_{\text{mix}}} \left\|\overline{\boldsymbol{K}}^{\ell,1}(t) - \overline{\boldsymbol{K}}^{\ell,2}(t)\right\| dt\right], \qquad (25)
\end{aligned}$$

where in the second inequality we have used the fact that f is 1-Lipschitz continuous under the L^1 norm of the space $\mathcal{K}^L$. For each pair of $\boldsymbol{k}, \boldsymbol{k}'$, observe that because $\overline{\sigma}$ is a $\boldsymbol{k}^0$-irreducible policy, $\mathbb{E}[\tau_{\text{mix}}]$ is finite; and because $\mathcal{K}$ is a finite set, $\left\|\overline{\boldsymbol{K}}^{\ell,1}(t) - \overline{\boldsymbol{K}}^{\ell,2}(t)\right\|$ is uniformly bounded. All these finite quantities depend on a single-server system under a policy $\overline{\sigma}$ that is independent of r. As a result, the last expression in (25) is of constant order. Moreover, because there are finite pairs of $(\boldsymbol{k}, \boldsymbol{k}')$, the supremum $\sup_{\boldsymbol{k},\boldsymbol{k}'} \mathbb{E}\left[\int_0^{\tau_{\text{mix}}} \|\overline{\boldsymbol{K}}^{\ell,1}(t) - \overline{\boldsymbol{K}}^{\ell,2}(t)\| dt\right]$ is also of constant order, independent of r. This proves the Stein factor bound in (23). Together with the generator comparison, we have proved

$$\sup_{f \in \text{Lip}(1)} \mathbb{E}\left[f\left(\overline{\boldsymbol{K}}^{1:L}\right)\right] - \mathbb{E}\left[f\left(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}\right)\right] = O\left(\sqrt{r}\right). \qquad (12)$$

Because $\sum_{\ell=1}^{L} \sum_{i \in \mathcal{I}} \eta_i^\ell \leq |\mathcal{I}|\, \eta_{\max} = O\left(\sqrt{r}\right)$, for any $f \in \text{Lip}(1)$, we have

$$\left|\mathbb{E}\left[f\left(\widehat{\boldsymbol{K}}^{1:L}\right)\right] - \mathbb{E}\left[f\left(\widehat{\boldsymbol{K}}^{1:L} + \boldsymbol{\eta}^{1:L}\right)\right]\right| \leq \mathbb{E}\left[\sum_{\ell=1}^{L} \sum_{i \in \mathcal{I}} \eta_i^\ell(t)\right] = O\left(\sqrt{r}\right). \qquad (26)$$

Plugging the above equation to (12), we get $\sup_{f \in \text{Lip}(1)} \mathbb{E}\left[f\left(\overline{\boldsymbol{K}}^{1:L}\right)\right] - \mathbb{E}\left[f\left(\widehat{\boldsymbol{K}}^{1:L}\right)\right] = O\left(\sqrt{r}\right)$. This proves Lemma 2. □

5.5 Role of Tokens and Virtual Jobs

This section aims to shed light on the role of tokens and virtual jobs in the proposed policy, JRS. We first outline how we devise the token-and-virtual-job mechanism from the perspective of generators. To begin with, consider the scenario where dispatch decisions are solely based on real-job configurations $(\boldsymbol{K}^\ell)_{\ell=1,2,\ldots}$. In this case, the transitions of servers' configurations would be correlated in general due to job arrivals, which are dispatched based on the joint configuration of all servers. To break this correlation, let us introduce tokens but not set an upper limit yet on the number of tokens (thus no virtual jobs). Observe that $\boldsymbol{K}^{1:L}(t) + \boldsymbol{\eta}^{1:L}(t)$ remains unchanged by

job arrivals, so tokens remove the correlation brought about by job arrivals. However, because tokens lack internal phase transitions or departures, the transition dynamics of $\boldsymbol{K}^{1:L}(t) + \boldsymbol{\eta}^{1:L}(t)$ will diverge from $\overline{\boldsymbol{K}}^{1:L}(t)$ when $\boldsymbol{\eta}(t)$ is large. In other words, although tokens help decouple the transitions on servers, they cannot keep the transitions of $\boldsymbol{K}^{1:L}(t) + \boldsymbol{\eta}^{1:L}(t)$ close to $\overline{\boldsymbol{K}}^{1:L}(t)$. To solve this issue, we finally introduce the mechanism of converting tokens into virtual jobs when the number of tokens is high, where virtual jobs can make internal phase transitions or departures just like real jobs. Now, the sum $\boldsymbol{K}^{1:L}(t) + \boldsymbol{\zeta}^{1:L}(t) + \boldsymbol{\eta}^{1:L}(t)$ remains unchanged by job arrivals nor the creation of virtual jobs, and the internal phase transitions and job departures are similar to those of $\overline{\boldsymbol{K}}^{1:L}(t)$. More formally, the generators of $\boldsymbol{K}^{1:L}(t) + \boldsymbol{\zeta}^{1:L}(t) + \boldsymbol{\eta}^{1:L}(t)$ and $\overline{\boldsymbol{K}}^{1:L}(t)$ are close to each other – their additive difference can be upper bounded by a quantity proportional to the expected number of tokens, as shown in (22). Therefore, by regulating the number of tokens, we can control the difference between the generators of $\boldsymbol{K}^{1:L}(t) + \boldsymbol{\zeta}^{1:L}(t) + \boldsymbol{\eta}^{1:L}(t)$ and $\overline{\boldsymbol{K}}^{1:L}(t)$.

Another key design component of JRS is that the subroutine requests jobs based on the observed configurations. This has been used in the proof of Lemma 2 to show that the observed configurations $\boldsymbol{K}^{1:L} + \boldsymbol{\zeta}^{1:L}$ are close to $\overline{\boldsymbol{K}}^{1:L}$, which consist of L i.i.d. single-server systems. Recall that each single-server system in $\overline{\boldsymbol{K}}^{1:L}$ is designed to have a throughput of $\lambda_i r/L$ for each job type $i \in \mathcal{I}$. Therefore, the proximity between $\boldsymbol{K}^{1:L} + \boldsymbol{\zeta}^{1:L}$ and $\overline{\boldsymbol{K}}^{1:L}$ ensures that the job request rate mirrors the arrival rate for each job type, regardless of the real-job configurations. The fact that these two rates are approximately equal is important for proving Lemma 3 and Lemma 4. It guarantees that both the rate of generating virtual jobs (when there are too many tokens) and the rate of dispatching jobs to backup servers (when there are no tokens) are appropriate.

A natural follow-up question is whether the usage of tokens and virtual jobs is fundamental or an artifact of our analysis technique. For example, it is unclear whether removing the upper limit on the number of tokens would still yield an asymptotically optimal policy. This is an interesting question that we do not have a complete answer to. The token-and-virtual-job mechanism emerges as a natural choice under our analysis framework. Nevertheless, it is worth noting that our analysis primarily treats each server's configuration as a generic Markov chain, without utilizing many properties specific to the stochastic bin-packing setting. An exception to this is the proofs in Appendix B.2, where we use the model that each job leaves the system within a constant expected time. It would be interesting to explore more properties of the problem to better understand policy designs without auxiliary state variables like tokens and virtual jobs.

6 CONCLUSION

In this paper, we study a new setting of stochastic bin-packing in service systems that features time-varying item sizes. Since our formulation is motivated by the problem of virtual-machine scheduling in computing systems, we use the terminology of jobs and servers, where jobs are viewed as items, whose sizes are their resource requirements, and servers as bins. The time-varying item sizes capture the emerging trend in practice that jobs' resource requirements vary over time. Our goal is to design a job dispatch policy to minimize the expected number of active servers in steady state, subject to a constraint on resource contentions. Our main result is the design of a policy that achieves an optimality gap of $O(\sqrt{r})$, where r is the scaling factor of the arrival rate. When specialized to the setting where jobs' resource requirements remain fixed over time, this result improves upon the state-of-the-art $o(r)$ optimality gap. Our technical approach highlights a novel policy conversion framework, Join-Requesting-Server, that reduces the policy design problem to that in a single-server system.

There are several potential directions that may be worth further exploration. One direction is to strengthen the optimality result within the current setting. Specifically, it is interesting to investigate: (i) whether it is possible to achieve an optimality gap smaller than $\Theta(\sqrt{r})$; and (ii) whether there exist asymptotically optimal policies whose average cost rate of resource contention satisfies the budget strictly instead of asymptotically.

We are also interested in extending our technique to the optimal control of other systems with similar structures. Intuitively, this technique could be applied to systems with many components that evolve mostly independently but are weakly coupled by certain constraints. Viewing each component as a server, we can define a suitable single-server problem and then design a policy for the original system to track the dynamics of the optimal single-server solution. Below we list several variations of our model that can potentially be analyzed using the proposed technique.

- A model where jobs running on each server will be put into a local queue when there are resource contentions. The goal thus becomes finding the optimal trade-offs between the number of active servers and the waiting time of the jobs.
- A model that allows each server to have a Markovian state that affects the dynamics of the jobs running on the server.
- A model that allows jobs to migrate to different servers at the cost of migration delays.
- A closed-system model where jobs re-enter the system after completion.

A third possible direction is to tackle the problem when the arrival rates and the parameters in the job model are unknown, as mentioned in Section 4.3. A possible approach is to develop an approximate version of the JRS framework, where the optimal single-server policy and the simulator for the virtual jobs are both learned from data. It is desirable to design such an approximate framework whose performance degrades gracefully as the approximation error increases.

ACKNOWLEDGMENTS

Y. Hong and W. Wang are supported in part by NSF grants CNS-200773 and ECCS-2145713. Q. Xie is supported in part by NSF grant CNS-1955997.

REFERENCES

[1] Nikhil Ayyadevara, Rajni Dabas, Arindam Khan, and K. V. N. Sreenivas. 2022. Near-Optimal Algorithms for Stochastic Online Bin Packing. In *Proc. Int. Conf. Automata, Languages and Programming (ICALP)*, Vol. 229. 12:1–12:20.

[2] Noman Bashir, Nan Deng, Krzysztof Rzadca, David Irwin, Sree Kodak, and Rohit Jnagal. 2021. Take It to the Limit: Peak Prediction-Driven Resource Overcommitment in Datacenters. In *Proc. European Conf. Computer Systems (EuroSys)*. Online Event, United Kingdom, 556–573.

[3] Anton Braverman. 2022. The Prelimit Generator Comparison Approach of Stein's Method. *Stoch. Syst.* 12, 2 (2022), 181–204.

[4] Anton Braverman and J. G. Dai. 2017. Stein's method for steady-state diffusion approximations of $M/Ph/n + M$ systems. *Ann. Appl. Probab.* 27 (Feb. 2017), 550–581.

[5] Anton Braverman, J. G. Dai, and Jiekun Feng. 2017. Stein's method for steady-state diffusion approximations: an introduction through the Erlang-A and Erlang-C models. *Stoch. Syst.* 6, 2 (2017), 301–366.

[6] Niv Buchbinder, Yaron Fairstein, Konstantina Mellou, Ishai Menache, and Joseph (Seffi) Naor. 2021. Online Virtual Machine Allocation with Lifetime and Load Predictions. *ACM SIGMETRICS Perform. Eval. Rev.* 49, 1 (May 2021), 9–10.

[7] Google Cloud. 2023. Overcommitting CPUs on sole-tenant VMs. https://cloud.google.com/compute/docs/nodes/overcommitting-cpus-sole-tenant-vms.

[8] Google Cloud. 2023. Virtual machine instances. https://cloud.google.com/compute/docs/instances.

[9] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. 1983. Dynamic Bin Packing. *SIAM J. Comput.* 12, 2 (1983), 227–258.

[10] Coastas Courcobetis and Richard Weber. 1990. Stability of On-Line Bin Packing with Random Arrivals and Long-Run-Average Constraints. *Probab. Eng. Inf. Sci.* 4, 4 (1990), 447–460.

[11] C. Courcoubetis and R. R. Weber. 1986. Necessary and Sufficient Conditions for Stability of a Bin-Packing System. *J. Appl. Probab.* 23, 4 (1986), 989–999.

[12] Janos Csirik, David S. Johnson, Claire Kenyon, James B. Orlin, Peter W. Shor, and Richard R. Weber. 2006. On the Sum-of-Squares Algorithm for Bin Packing. *J. ACM* 53, 1 (Jan. 2006), 1–65.
[13] Christina Delimitrou and Christos Kozyrakis. 2014. Quasar: Resource-Efficient and QoS-Aware Cluster Management. In *Proc. Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. Salt Lake City, UT, 127–144.
[14] Apache Software Foundation. 2023. Apache Mesos: Containerizers. https://mesos.apache.org/documentation/latest/containerizers/.
[15] Apache Software Foundation. 2023. Apache Mesos: Oversubscription. https://mesos.apache.org/documentation/latest/oversubscription/.
[16] Daniel Freund and Siddhartha Banerjee. 2019. Good prophets know when the end is near. *Available at SSRN: https://ssrn.com/abstract=3479189* (Nov. 2019).
[17] Javad Ghaderi, Yuan Zhong, and R. Srikant. 2014. Asymptotic Optimality of BestFit for Stochastic Bin Packing. *SIGMETRICS Perform. Eval. Rev.* 42, 2 (Sept. 2014), 64–66.
[18] Varun Gupta and Ana Radovanović. 2020. Interior-Point-Based Online Stochastic Bin Packing. *Oper. Res.* 68, 5 (2020), 1474–1492.
[19] Leonard Kleinrock. 1975. *Queueing Systems*. John Wiley & Son.
[20] Yusen Li, Xueyan Tang, and Wentong Cai. 2014. On Dynamic Bin Packing for Resource Allocation in the Cloud. In *Proc. Ann. ACM Symp. Parallelism in Algorithms and Architectures (SPAA)*. Prague, Czech Republic, 2–11.
[21] David Lo, Liqun Cheng, Rama Govindaraju, Parthasarathy Ranganathan, and Christos Kozyrakis. 2015. Heracles: Improving resource efficiency at scale. In *Proc. ACM/IEEE Ann. Int. Symp. Computer Architecture (ISCA)*. Portland, OR, 450–462.
[22] Siva Theja Maguluri and R. Srikant. 2013. Scheduling jobs with unknown duration in clouds. In *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*. Turin, Italy, 1887–1895.
[23] Siva Theja Maguluri, R Srikant, and Lei Ying. 2012. Stochastic Models of Load Balancing and Scheduling in Cloud Computing Clusters. In *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*. Orlando, FL, 702–710.
[24] Siva Theja Maguluri, R. Srikant, and Lei Ying. 2014. Heavy traffic optimal resource allocation algorithms for cloud computing clusters. *Perform. Eval.* 81 (2014), 20–39.
[25] Sean Meyn. 2007. *Control Techniques for Complex Networks* (1st ed.). Cambridge University Press, USA.
[26] Konstantinos Psychas and Javad Ghaderi. 2018. On Non-Preemptive VM Scheduling in the Cloud. In *Proc. ACM SIGMETRICS Int. Conf. Measurement and Modeling of Computer Systems*. Irvine, CA, 67–69.
[27] Konstantinos Psychas and Javad Ghaderi. 2019. Scheduling Jobs with Random Resource Requirements in Computing Clusters. In *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*. 2269–2277.
[28] Konstantinos Psychas and Javad Ghaderi. 2021. High-Throughput Bin Packing: Scheduling Jobs With Random Resource Demands in Clusters. *IEEE/ACM Trans. Netw.* 29, 1 (2021), 220–233.
[29] Konstantinos Psychas and Javad Ghaderi. 2022. A Theory of Auto-Scaling for Resource Reservation in Cloud Services. *Stoch. Syst.* 12, 3 (2022), 227–252.
[30] Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Randy H. Katz, and Michael A. Kozuch. 2012. Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis. In *Proc. ACM Symp. Cloud Computing (SoCC)*. San Jose, CA, Article 7, 13 pages.
[31] Krzysztof Rzadca, Pawel Findeisen, Jacek Swiderski, Przemyslaw Zych, Przemyslaw Broniek, Jarek Kusmierek, Pawel Nowak, Beata Strack, Piotr Witusowski, Steven Hand, and John Wilkes. 2020. Autopilot: Workload Autoscaling at Google. In *Proc. European Conf. Computer Systems (EuroSys)*. Heraklion, Greece, Article 16, 16 pages.
[32] Alexander L. Stolyar. 2013. An Infinite Server System with General Packing Constraints. *Oper. Res.* 61, 5 (2013), 1200–1217.
[33] Alexander L. Stolyar. 2017. Large-scale heterogeneous service systems with general packing constraints. *Adv. Appl. Probab.* 49 (March 2017), 61–83. Issue 1.
[34] Alexander L. Stolyar and Yuan Zhong. 2013. A large-scale service system with packing constraints: Minimizing the number of occupied servers. *ACM SIGMETRICS Perform. Eval. Rev.* 41, 1 (June 2013), 41–52.
[35] Alexander L. Stolyar and Yuan Zhong. 2015. Asymptotic optimality of a greedy randomized algorithm in a large-scale service system with general packing constraints. *Queueing Syst.* 79 (June 2015), 117–143. Issue 2.
[36] Alexander L. Stolyar and Yuan Zhong. 2021. A Service System with Packing Constraints: Greedy Randomized Algorithm Achieving Sublinear in Scale Optimality Gap. *Stoch. Syst.* 11 (June 2021), 83–111. Issue 2.
[37] Muhammad Tirmazi, Adam Barker, Nan Deng, Md E. Haque, Zhijing Gene Qin, Steven Hand, Mor Harchol-Balter, and John Wilkes. 2020. Borg: The next Generation. In *Proc. European Conf. Computer Systems (EuroSys)*. Heraklion, Greece, Article 30, 14 pages.
[38] John Wilkes. 2019. Google cluster-usage traces v3. http://github.com/google/cluster-data.

[39] Qiaomin Xie, Xiaobo Dong, Yi Lu, and R. Srikant. 2015. Power of d Choices for Large-Scale Bin Packing: A Loss Model. In *Proc. ACM SIGMETRICS Int. Conf. Measurement and Modeling of Computer Systems*. Portland, OR, 321–334.

A PROOF OF THEOREM 2 (LOWER BOUND)

Proof. It is sufficient to show that given an infinite-server policy σ for $\mathcal{P}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ in (2), we have $N(\sigma) \geq \overline{N}^*$. To this end, we will construct a single-server policy $\overline{\sigma}$ such that the resulting system configuration $\overline{\boldsymbol{K}}(\infty)$ in steady state satisfies:

$$\mathbb{E}\left[h\big(\overline{\boldsymbol{K}}(\infty)\big)\big|\overline{\boldsymbol{K}}(\infty) \neq \mathbf{0}\right] = C(\sigma) \leq \epsilon, \tag{27}$$

$$\overline{\lambda}_i = \frac{\lambda_i r}{N(\sigma)}, \quad \forall i \in \mathcal{I}. \tag{28}$$

Let π be the distribution of $\overline{\boldsymbol{K}}(\infty)$, then $(N(\sigma), \overline{\sigma}, \pi)$ is a feasible solution to the problem $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ in (3). As a result, we have $N(\sigma) \geq \overline{N}^*$. Note that although $\overline{\sigma}$ is actually non-Markovian, i.e., it makes decisions based on not only the current configuration but also the history, as we will show in Appendix C, $\overline{N}^*$ is still a lower bound to the objective value that $\overline{\sigma}$ can achieve in $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$.

The construction of the single-server policy $\overline{\sigma}$ involves simulating an infinite-server system under σ from the empty configuration. At time 0, the policy $\overline{\sigma}$ randomly chooses the ℓ-th server in the infinite-server system with probability p^ℓ, for $\ell = 1, 2, \cdots$. It then requests jobs for the single-server system according to a policy $\overline{\sigma}^\ell$. The key to our policy $\overline{\sigma}^\ell$ is to make the single-server system emulate the job assignment at the ℓ-th server of the simulated infinite-server system, but without incurring idleness. We first construct the policy $\overline{\sigma}^\ell$, and then specify the probabilities p^ℓ.

Let us start by introducing some useful notation. Let $\overline{\boldsymbol{K}}^\ell(t)$ be the single-server system configuration under $\overline{\sigma}^\ell$ at time t and $\boldsymbol{K}^\ell(t)$ be the configuration of the ℓ-th simulated server in the infinite-server system under σ. We define a stochastic process $\{s^\ell(t), t \geq 0\}$ as follows: $s^\ell(t) = \max_\tau \left\{\tau : \int_0^\tau \mathbb{1}_{\{\boldsymbol{K}^\ell(x)\neq\mathbf{0}\}} dx = t\right\}$. The "max" is well-defined because the integral is continuous in τ. Intuitively, $s^\ell(t)$ gives the maximum time when the accumulative busy time of the ℓ-th server is t. Note that $\{s^\ell(t), t \geq 0\}$ is only discontinuous when $\boldsymbol{K}^\ell(\tau)$ reaches 0, thus it is right-differentiable with derivative equal to 1 at any point.

We construct $\overline{\sigma}^\ell$ and the simulation of the infinite-server system under σ in a way such that:

$$\overline{\boldsymbol{K}}^\ell(t) = \boldsymbol{K}^\ell(s^\ell(t)) \quad \forall t. \tag{29}$$

That is, we want that the single-server system has the same dynamic of the simulated ℓ-th server except skipping the idle period. To this end, we couple the two systems as follows:

(1) When the ℓ-th simulated server $\boldsymbol{K}^\ell(s^\ell(t))$ receives a type i job, we let the single-server system $\overline{\boldsymbol{K}}^\ell(t)$ request a type i job at time t. For each such job, its phase transition process in the ℓ-th simulated server is the same as that in the single-server system. That is, when we observe any internal transition or departure event in $\overline{\boldsymbol{K}}^\ell(t)$, we produce a same event on the ℓ-th simulated server $\boldsymbol{K}^\ell(s^\ell(t))$.
(2) The simulations of the rest of the infinite-server system under policy σ are driven by independently generated random seeds.

It is not hard to see that the simulated infinite server-system has the same stochastic behavior as an uncoupled system under σ. Moreover, as we couple all the events that happen in $\overline{\boldsymbol{K}}^\ell(t)$ and $\boldsymbol{K}^\ell(s^\ell(t))$, together with the facts that $\overline{\boldsymbol{K}}^\ell(t)$ and $\boldsymbol{K}^\ell(s^\ell(t))$ are piecewise constant and $\overline{\boldsymbol{K}}^\ell(0-) = \boldsymbol{K}^\ell(s^\ell(0-)) = \mathbf{0}$, we get (29).

Next we claim that (29) implies the following relationship between the steady-state cost of the single-server system under $\overline{\sigma}^\ell$ and the steady-state cost of the ℓ-th simulated server in the

infinite-server system under σ:

$$\mathbb{E}\left[h\left(\overline{\boldsymbol{K}}^{\ell}(\infty)\right)\right] = \frac{\mathbb{E}\left[h(\boldsymbol{K}^{\ell}(\infty))\right]}{\mathbb{P}\left(\boldsymbol{K}^{\ell}(\infty) \neq \boldsymbol{0}\right)}. \tag{30}$$

This is because for all $\boldsymbol{k} \neq \boldsymbol{0}$, we have

$$\begin{aligned}\frac{\mathbb{P}\left(\boldsymbol{K}^{\ell}(\infty) = \boldsymbol{k}\right)}{\mathbb{P}\left(\boldsymbol{K}^{\ell}(\infty) \neq \boldsymbol{0}\right)} &\overset{(a)}{=} \lim_{S\to\infty} \frac{\int_0^S \mathbb{1}_{\{\boldsymbol{K}^{\ell}(s)=\boldsymbol{k}\}} ds}{\int_0^S \mathbb{1}_{\{\boldsymbol{K}^{\ell}(s)\neq\boldsymbol{0}\}} ds} \overset{(b)}{=} \lim_{T\to\infty} \frac{\int_0^T \mathbb{1}_{\{\boldsymbol{K}^{\ell}(s^{\ell}(t))=\boldsymbol{k}\}} dt}{\int_0^T \mathbb{1}_{\{\boldsymbol{K}^{\ell}(s^{\ell}(t))\neq\boldsymbol{0}\}} dt} \\ &\overset{(c)}{=} \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{1}_{\{\boldsymbol{K}^{\ell}(s^{\ell}(t))=\boldsymbol{k}\}} dt \overset{(d)}{=} \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{1}_{\{\overline{\boldsymbol{K}}^{\ell}(t)=\boldsymbol{k}\}} dt \\ &\overset{(e)}{=} \mathbb{P}\left(\overline{\boldsymbol{K}}^{\ell}(\infty) = \boldsymbol{k}\right),\end{aligned}$$

where (a) and (e) hold because long-run averages converge to steady-state expectations; (b) is due to the fact that $\int_0^T \mathbb{1}_{\{\boldsymbol{K}^{\ell}(s^{\ell}(t))=\boldsymbol{k}\}} dt = \int_0^{s^{\ell}(T)} \mathbb{1}_{\{\boldsymbol{K}^{\ell}(s)=\boldsymbol{k}\}} ds$, for any $\boldsymbol{k} \neq \boldsymbol{0}$; (c) is due to the fact that $\mathbb{1}_{\{\boldsymbol{K}^{\ell}(s^{\ell}(t))\neq\boldsymbol{0}\}} = 1$; and (d) follows from (29).

Let $\overline{\lambda}_i^{\ell}$ be the long-run request rates of type i jobs in the single-server system under $\overline{\sigma}^{\ell}$, and λ_i^{ℓ} be the throughput of type i jobs of ℓ-th simulated server under σ. By the construction of $\overline{\sigma}^{\ell}$, the single-server system requests jobs based on the arrival events of the ℓ-th simulated server, we have $\overline{\lambda}_i^{\ell} = \frac{\lambda_i^{\ell}}{\mathbb{P}(\boldsymbol{K}^{\ell}(\infty)\neq\boldsymbol{0})}, \forall i \in \mathcal{I}$.

With the constructed policies $\{\overline{\sigma}^{\ell}, \ell = 1, 2, \dots\}$, we are ready to define the policy $\overline{\sigma}$. We let $\overline{\sigma}$ choose an index ℓ with probability p^{ℓ} at time 0, and then follow $\overline{\sigma}^{\ell}$. We set the probability p^{ℓ} as

$$p^{\ell} = \frac{\mathbb{P}\left(\boldsymbol{K}^{\ell}(\infty) \neq \boldsymbol{0}\right)}{\sum_{\ell'=1}^{\infty} \mathbb{P}\left(\boldsymbol{K}^{\ell'}(\infty) \neq \boldsymbol{0}\right)} = \frac{\mathbb{P}\left(\boldsymbol{K}^{\ell}(\infty) \neq \boldsymbol{0}\right)}{\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}(\infty)]}, \quad \forall \ell = 1, 2, \dots \tag{31}$$

where the second inequality uses the fact that $\sum_{\ell'=1}^{\infty} \mathbb{P}\left(\boldsymbol{K}^{\ell'}(\infty) \neq \boldsymbol{0}\right) = \sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}(\infty)]$. Then under $\overline{\sigma}$, we have

$$\begin{aligned}\mathbb{E}\left[h\left(\overline{\boldsymbol{K}}(\infty)\right)\right] &= \sum_{\ell=1}^{\infty} p^{\ell} \mathbb{E}\left[h\left(\overline{\boldsymbol{K}}^{\ell}(\infty)\right)\right] \overset{(a)}{=} \sum_{\ell=1}^{\infty} \frac{\mathbb{P}\left(\boldsymbol{K}^{\ell}(\infty) \neq \boldsymbol{0}\right)}{\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}(\infty)]} \cdot \frac{\mathbb{E}\left[h(\boldsymbol{K}^{\ell}(\infty))\right]}{\mathbb{P}\left(\boldsymbol{K}^{\ell}(\infty) \neq \boldsymbol{0}\right)} \\ &= \frac{\sum_{\ell=1}^{\infty} \mathbb{E}\left[h(\boldsymbol{K}^{\ell}(\infty))\right]}{\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}(\infty)]} = \frac{\sum_{\boldsymbol{k}\neq\boldsymbol{0}} h(\boldsymbol{k}) \mathbb{E}[X_{\boldsymbol{k}}(\infty)]}{\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}(\infty)]} = C(\sigma),\end{aligned}$$

where (a) follows from (30) and (31). Observe that under $\overline{\sigma}$, $\overline{\boldsymbol{K}}(\infty) \neq \boldsymbol{0}$ almost surely, we thus have

$$\mathbb{E}\left[h\left(\overline{\boldsymbol{K}}(\infty)\right) \middle| \overline{\boldsymbol{K}}(\infty) \neq \boldsymbol{0}\right] = \mathbb{E}\left[h\left(\overline{\boldsymbol{K}}(\infty)\right)\right] = C(\sigma),$$

which proves (27). Moreover, for each $i \in \mathcal{I}$ the request rate $\overline{\lambda}_i$ is given by

$$\overline{\lambda}_i = \sum_{\ell=1}^{\infty} p^{\ell} \cdot \overline{\lambda}_i^{\ell} = \sum_{\ell=1}^{\infty} \frac{\mathbb{P}\left(\boldsymbol{K}^{\ell}(\infty) \neq \boldsymbol{0}\right)}{\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}(\infty)]} \cdot \frac{\lambda_i^{\ell}}{\mathbb{P}\left(\boldsymbol{K}^{\ell}(\infty) \neq \boldsymbol{0}\right)} = \frac{\sum_{\ell=1}^{\infty} \lambda_i^{\ell}}{\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}(\infty)]} = \frac{\lambda_i r}{N(\sigma)}.$$

This proves (28). By the argument presented at the beginning of the proof, we get $N(\sigma) \geq \overline{N}^*$. □

B THE REST OF THE PROOFS NEEDED FOR THEOREM 3 (CONVERSION THEOREM)

B.1 Proof of Lemma 1

Proof. We will show that under the JRS policy, the Markov chain for the system state (represented as $((\boldsymbol{K}^{\ell}(t))_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$) has a unique stationary distribution by first arguing that it is $\boldsymbol{k}^0$-irreducible (here being $\boldsymbol{k}^0$-irreducible means the Markov chain has a state that can be reached

by all other states through transitions, "$\boldsymbol{k}^0$" in "$\boldsymbol{k}^0$-irreducible" does not refer to any specific states), and then use Foster-Lyapunov theorem to show the positive recurrence [see, e.g., 25]. Combining $\boldsymbol{k}^0$-irreducibility with positive recurrence, we can conclude that the Markov chain under study has a unique stationary distribution.

First, we show that the Markov chain $((\boldsymbol{K}^\ell(t))_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$ is $\boldsymbol{k}^0$-irreducible. Specifically, observe that the Markov chain starting from any state $((\boldsymbol{k}^\ell)_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}, \boldsymbol{\eta}^{1:L})$ can reach the state $(\boldsymbol{k}^{1:L}, (\boldsymbol{0})_{\ell=L+1,\dots}, \boldsymbol{0}^{1:L}, \boldsymbol{0}^{1:L})$, after experiencing a sequence of departures and arrivals that clears up all the tokens, virtual jobs and jobs on backup servers. Further, letting $\widetilde{\boldsymbol{k}}$ be the configuration reachable by all other configuration in the single-server system under the policy $\overline{\sigma}$, we argue that starting from any states of the form $(\boldsymbol{k}^{1:L}, (\boldsymbol{0})_{\ell=L+1,\dots}, \boldsymbol{0}^{1:L}, \boldsymbol{0}^{1:L})$, the Markov chain can reach the state $(\widetilde{\boldsymbol{k}}^{1:L}, (\boldsymbol{0})_{\ell=L+1,\dots}, \boldsymbol{0}^{1:L}, \boldsymbol{0}^{1:L})$. Because for any $\ell \le L$, there is a transition path from $\boldsymbol{k}^\ell$ to $\widetilde{\boldsymbol{k}}$, consider the sequence of events where each $\boldsymbol{K}^\ell(t)$ transitions independently following the path, and the jobs arrive right after $\boldsymbol{K}^\ell(t)$ making a request, so that the tokens are checked out before $\boldsymbol{K}^\ell(t)$ has a further transition. In this way, each $\boldsymbol{K}^\ell(t)$ with $\ell \le L$ can eventually reach $\widetilde{\boldsymbol{k}}$ from $\boldsymbol{k}^\ell$. This proves the $\boldsymbol{k}^0$-irreducibility of $((\boldsymbol{K}^\ell(t))_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$.

Next, we show that $((\boldsymbol{K}^\ell(t))_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$ satisfies the Foster-Lyapunov criterion, i.e.,

$$\widehat{G}g \le -1 + b\mathbb{1}_{\{S\}}, \tag{32}$$

where g is a non-negative function of the states, S is a finite set, b is a finite number, and $\widehat{G}$ is the infinitesimal generator of the continuous-time Markov chain. Let t_i be the expected remaining time in the system when a job is in phase i for each $i \in \mathcal{I}$. According to the job model, we have the recurrence relation

$$\Big(\mu_{i\perp} + \sum_{i' \in \mathcal{I}\,:\, i' \neq i} \mu_{ii'}\Big) t_i = \sum_{i' \in \mathcal{I}\,:\, i' \neq i} \mu_{ii'} t_{i'} \quad \forall i \in \mathcal{I}. \tag{33}$$

We construct a Lyapunov function g as follows:

$$g((\boldsymbol{k}^\ell)_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}, \boldsymbol{\eta}^{1:L}) = \sum_{i \in \mathcal{I}} \sum_{\ell=1}^{\infty} t_i k_i^\ell + \sum_{i \in \mathcal{I}} \sum_{\ell=1}^{\infty} t_i \zeta_i^\ell. \tag{34}$$

Using the relation (33), it can be verified that the drift of g satisfies

$$\begin{aligned}
&\widehat{G}g((\boldsymbol{k}^\ell)_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}, \boldsymbol{\eta}^{1:L}) \\
\le &\sum_{i \in \mathcal{I}} \Big(\lambda_i t_i r - \sum_{\ell=1}^{\infty} k_i^\ell\Big) + \sum_{i \in \mathcal{I}} \Big(\sum_{\ell=1}^{L} \sum_{(\boldsymbol{k}', \boldsymbol{a}) \in E(\boldsymbol{k}^\ell)} \gamma_{\boldsymbol{k}^\ell, (\boldsymbol{k}', \boldsymbol{a})} a_i t_i - \sum_{\ell=1}^{\infty} \zeta_i^\ell \Big) \\
\le &\sum_{i \in \mathcal{I}} \Big(\lambda_i t_i r + L \cdot \max_{\boldsymbol{k} \in \mathcal{K}} \sum_{(\boldsymbol{k}', \boldsymbol{a}) \in E(\boldsymbol{k})} \gamma_{\boldsymbol{k}, (\boldsymbol{k}', \boldsymbol{a})} a_i t_i\Big) - \Big(\sum_{\ell=1}^{\infty} \sum_{i \in \mathcal{I}} k_i^\ell + \sum_{\ell=1}^{\infty} \sum_{i \in \mathcal{I}} \zeta_i^\ell \Big),
\end{aligned}$$

where the first inequality uses the fact that virtual jobs are generated at a rate no faster than the total rate of job requests. Then the Foster-Lyapunov criterion in (32) is satisfied with b and S given by

$$b = \sum_{i \in \mathcal{I}} \Big(\lambda_i t_i r + L \cdot \max_{\boldsymbol{k} \in \mathcal{K}} \sum_{(\boldsymbol{k}', \boldsymbol{a}) \in E(\boldsymbol{k})} \gamma_{\boldsymbol{k}, (\boldsymbol{k}', \boldsymbol{a})} a_i t_i\Big),$$

$$S = \left\{((\boldsymbol{k}^\ell)_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}, \boldsymbol{\eta}^{1:L}) : g((\boldsymbol{k}^\ell)_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}, \boldsymbol{\eta}^{1:L}) \le b + 1\right\}.$$

By the Foster-Lyapunov theorem, $((\boldsymbol{K}^\ell(t))_{\ell=1,2,\dots}, \boldsymbol{\zeta}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$ is positive recurrent. □

B.2 Proofs of Lemma 3 and Lemma 4

In this subsection, we prove Lemma 3 and Lemma 4 together. We begin by introducing some notations. We use $\boldsymbol{U}^\ell \triangleq (\widehat{\boldsymbol{K}}^\ell, \boldsymbol{\eta}^\ell)$ to represent the state of the ℓ-th server, and use $\boldsymbol{U}^{1:L}$ to represent the joint state of the first L servers. We also use the lowercase $\boldsymbol{u}^\ell$, $\boldsymbol{u}^{1:L}$ to represent the realizations of the corresponding random variables. We denote the total number of type i virtual jobs as $V_i \triangleq \sum_{\ell=1}^L \zeta_i^\ell$ for $i \in \mathcal{I}$, and its realization as v_i. We denote the total number of type i jobs on backup servers as Y_i for $\in \mathcal{I}$, and its realizations as y_i. We also denote the total number of type i tokens throughout the system as $Z_i \triangleq \sum_{\ell=1}^L \eta_i^\ell$, and its realization as z_i. Our goal can be rewritten as proving $\mathbb{E}[V_i] = O\left(\sqrt{r}\right)$ and $\mathbb{E}[Y_i] = O\left(\sqrt{r}\right)$ for each $i \in \mathcal{I}$.

We first give an overview of the proof. Observe that in our model, the expected time that a job stay in the system is fixed. As a result, bounding the number of virtual jobs or jobs on backup servers in the system is equivalent to bounding the rate that they are generated, according to Little's Law. By our construction of the policy, the rate of generating those jobs is closely related to the dynamics of the total number of type i tokens $Z_i(t)$.

To describe the dynamics of $Z_i(t)$, we first introduce two functions dv_i and dy_i:

$$dv_i(a_i, z_i) \triangleq (z_i + a_i - \eta_{\max})^+,$$
$$dy_i(z_i) \triangleq (1 - z_i)^+.$$

The function dv_i represents the increment in the number of type i virtual jobs due to the event that the total number of type i tokens on the first L servers exceeds the token limit $\eta_{\max}$. The function dy_i corresponds to the increment in the total number of type i jobs on backup servers due to the event that a type i job arrives to the system without seeing a type i token. For a function $g\colon (\mathcal{K}\times\mathcal{K})^L \to \mathbb{R}$ that only depends on the number of type i tokens z_i, its drift can be written as

$$\begin{aligned}\widehat{G}g(\boldsymbol{u}^{1:L}) = &\sum_{\ell=1}^L \sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\boldsymbol{k}^\ell)} \gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(g(z_i + a_i - dv_i(a_i,z_i)) - g(z_i)\right) \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} \\ &+ \lambda_i r \left(g(z_i - 1 + dy_i(z_i)) - g(z_i)\right). \end{aligned} \tag{35}$$

We abuse the notation of g here. For ease of exposition, we will simply write dv_i and dy_i to represent $dv_i(a_i, z_i)$ and $dy_i(z_i)$.

By construction, the total number of type i tokens $\{Z_i(t)\}$ is a stochastic process constrained within $[0, \eta_{\max}]$. Note that $Z_i(t)$ increases when some servers request new tokens, and decreases when a real or virtual arrival checks out the token or when some servers have the excessive tokens removed. When $Z_i(t)$ is away from the boundaries, the average rate that it increases is approximately equal to $\lambda_i r$, and the average rate that $Z_i(t)$ decreases is given by

$$\mathbb{E}\left[\sum_{\ell=1}^L \sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\widehat{\boldsymbol{K}}^\ell)} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right] \approx \mathbb{E}\left[\sum_{\ell=1}^L \sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\overline{\boldsymbol{K}}^\ell)} \gamma_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i\right] = L\cdot\overline{\lambda}_i \approx \lambda_i r,$$

where we have used the approximations that $\widehat{\boldsymbol{K}}^\ell \overset{d}{\approx} \overline{\boldsymbol{K}}^\ell$, $\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} \approx 1$ and $L = \lceil \overline{N} \rceil \approx \overline{N}$.

As $\{Z_i(t)\}$ randomly moves up and down with approximately the same rate and reflects on the boundaries of 0 and $\eta_{\max}$, it behaves as a reflected simple symmetric random walk. Intuitively speaking, the steady-state distribution of Z_i is approximately a uniform distribution over $[0, \eta_{\max}]$. Recall that dv_i and dy_i can only be non-zero when $Z_i(t)$ is near the boundaries. Since the length of the interval $\eta_{\max} = \Theta\left(\sqrt{r}\right)$, we can expect that dv_i and dy_i diminish as $r \to \infty$.

In the proof, we first establish the relationship between $\mathbb{E}[V_i]$, $\mathbb{E}[Y_i]$ and dv_i, dy_i using Little's Law. Then we derive bounds on dv_i and dy_i by analyzing the drift of several test functions of

Z_i. This step is implicitly based on the intuition that Z_i is approximately uniformly distributed over $[0, \eta_{\max}]$, with the tokens being generated and eliminated at similar rates. Finally, we invoke Lemma 2 to show that the tokens are indeed generated and eliminated at similar speeds, which leads to bounds on dv_i and dy_i.

Finally, we make some additional remarks on the notations. First, dv_i and dy_i depend on the total number of type i tokens z_i and the number of newly requested jobs a_i, although we omit the dependency expression for ease of exposition. Second, we abuse the notation dv_i and dy_i to denote the corresponding random variables. We also write $\sum_{\boldsymbol{k}',\boldsymbol{a}}$ as a shorthand for $\sum_{(\boldsymbol{k}',\boldsymbol{a})\in E(\boldsymbol{k}^\ell)}$ when the context is clear.

B.2.1 *Proofs of Lemma 3 and Lemma 4.*

We are now ready to prove Lemma 3 and Lemma 4.

Proof. **Step 1: Bounding Virtual Jobs and Jobs on Backup Servers using Little's Law.** We first apply Little's Law to V_i and Y_i. For each type i, we let the expected time that a type i job stays in the system be t_i. Let $t_{\max} = \max_{i\in\mathcal{I}} t_i$. Because there are only finitely many types of jobs, and each job spends finite expected time in the system, $t_{\max}$ is a finite constant. By Little's law,

$$\mathbb{E}[V_i] \le t_{\max}\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} dv_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right], \tag{36}$$

$$\mathbb{E}[Y_i] \le t_{\max}\mathbb{E}\left[\lambda_i r \cdot dy_i\right]. \tag{37}$$

Step 2: Drift Analysis. The above two equations (36) and (37) suggest that we can derive upper bounds on $\mathbb{E}[V_i]$ and $\mathbb{E}[Y_i]$ by analyzing the following two terms:

- $\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} dv_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right]$, interpreted as the average rate that $Z_i(t)$ reflects on the boundary at $\eta_{\max}$;
- $\mathbb{E}\left[\lambda_i r \cdot dy_i\right]$, interpreted as the average rate that $Z_i(t)$ reflects on the boundary at 0.

We establish the relationships of dv_i, dy_i and Z_i by analyzing the drift of two test functions g.

Letting $g(z_i) = z_i$ and taking steady-state expectation over its drift, by (35) and the fact that the drift is zero in steady state, we get

$$\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} (a_i - dv_i)\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} + \lambda_i r(-1 + dy_i)\right] = 0. \tag{38}$$

Similarly, letting $g(z_i) = z_i^2$ and taking steady-state expectation over its drift, one can verify that

$$\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} dv_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right] \tag{39}$$

$$= \frac{1}{\eta_{\max}} \cdot \mathbb{E}\left[\left(\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} - \lambda_i r\right) \cdot Z_i\right] \tag{40}$$

$$+ \frac{1}{2\eta_{\max}} \cdot \mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(a_i^2 - (dv_i)^2\right) \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} + \lambda_i r \cdot (1 - (dy_i)^2)\right]. \tag{41}$$

Readers may refer to the complete calculation at the end of this subsection.

Step 3: Estimating the Terms Obtained from Drift Analysis. We will first focus on bounding $\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} dv_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right]$ analyzing the two terms in (40) and (41) separately. Then we invoke (38) to bound $\mathbb{E}\left[\lambda_i r \cdot dy_i\right]$.

The term in (41) is easy to deal with. Observe that the number of jobs requested each time should be no more than the maximal number of jobs that a server can hold, i.e., $a_i \le K_{\max}$, so

$$\begin{aligned}(41) &\le \frac{1}{2\eta_{\max}} \cdot \mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} K_{\max}^2 + \lambda_i r\right] \\ &\le \frac{1}{2\eta_{\max}} \cdot \mathbb{E}\left[\sum_{\ell=1}^{L} \gamma_{\max} K_{\max}^2 + \lambda_i r\right] \\ &= O\left(\sqrt{r}\right), \end{aligned} \tag{42}$$

where in the second inequality we have used the fact that the total rate is uniformly bounded by $\gamma_{\max}$, and the last step uses the facts that $L = O(r)$ and $\eta_{\max} = \Theta\left(\sqrt{r}\right)$.

To bound the term in (40), first observe that $Z_i \le \eta_{\max}$, which implies that

$$(40) \le \mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} - \lambda_i r\right|\right].$$

The term on the RHS of the above equation is the expected absolute difference between the rates of generating and eliminating type i tokens, which can be shown to be small relative to r. Specifically, we claim that

$$\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} - \lambda_i r\right|\right] = O\left(\sqrt{r}\right). \tag{43}$$

To show (43), first notice that we can remove the indicator $\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}$ without introducing much error:

$$\begin{aligned}&\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} - \lambda_i r\right|\right] \\ \le\ &\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right|\right] + \mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell\neq\boldsymbol{0}\}}\right|\right] \\ \le\ &\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right|\right] + \mathbb{E}\left[\left|\sum_{\ell=1}^{L} \gamma_{\max} K_{\max} \mathbb{1}_{\{\boldsymbol{\eta}^\ell\neq\boldsymbol{0}\}}\right|\right] \\ \le\ &\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right|\right] + \gamma_{\max} K_{\max} |\mathcal{I}|\, \eta_{\max}, \end{aligned}$$

where the first inequality is due to triangular inequality, the second inequality is due to the definition of $\gamma_{\max}$, and the last inequality is because $\sum_{\ell=1}^{L} \mathbb{1}_{\{\boldsymbol{\eta}^\ell\neq\boldsymbol{0}\}} \le |\mathcal{I}|\, \eta_{\max}$. It remains to bound the term $\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right|\right]$, which can be seen as showing that the rate of generating type i tokens concentrates around the type i jobs' arrival rate $\lambda_i r$. It is natural to think of using some Law of Large Numbers. Unfortunately, $\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i$ is not a sum of i.i.d. random variables due to dependencies among $\widehat{\boldsymbol{K}}^\ell$ for different ℓ's. As a result, we want to invoke the Wasserstein distance bound in Lemma 2 to replace $\widehat{\boldsymbol{K}}^\ell$ in the above expression with $\overline{\boldsymbol{K}}^\ell$. We define the function $f(\boldsymbol{k}^{1:L})$ as

$$f(\boldsymbol{k}^{1:L}) = \frac{1}{2\gamma_{\max} K_{\max}} \left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right|. \tag{44}$$

We claim that $f \in \mathrm{Lip}(1)$. For any two $\boldsymbol{k}^{1:L,1}, \boldsymbol{k}^{1:L,2}$,

$$\begin{aligned}
&2\gamma_{\max}K_{\max} \cdot \left(f(\boldsymbol{k}^{1:L,1}) - f(\boldsymbol{k}^{1:L,2})\right) \\
&= \left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\boldsymbol{k}^{\ell,1},(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right| - \left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\boldsymbol{k}^{\ell,2},(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right| \\
&\le \left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\boldsymbol{k}^{\ell,1},(\boldsymbol{k}',\boldsymbol{a})} a_i - \sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\boldsymbol{k}^{\ell,2},(\boldsymbol{k}',\boldsymbol{a})} a_i\right| \\
&= \left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \left(\gamma_{\boldsymbol{k}^{\ell,1},(\boldsymbol{k}',\boldsymbol{a})} - \gamma_{\boldsymbol{k}^{\ell,2},(\boldsymbol{k}',\boldsymbol{a})}\right) a_i \mathbb{1}_{\{(\boldsymbol{k}^{\ell,1}) \neq (\boldsymbol{k}^{\ell,2})\}}\right| \\
&\le \sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \left|\gamma_{\boldsymbol{k}^{\ell,1},(\boldsymbol{k}',\boldsymbol{a})} - \gamma_{\boldsymbol{k}^{\ell,2},(\boldsymbol{k}',\boldsymbol{a})}\right| \cdot K_{\max} \cdot \|\boldsymbol{k}^{\ell,1} - \boldsymbol{k}^{\ell,2}\| \\
&\le \sum_{\ell=1}^{L} 2\gamma_{\max}K_{\max} \cdot \|\boldsymbol{k}^{\ell,1} - \boldsymbol{k}^{\ell,2}\| \\
&= 2\gamma_{\max}K_{\max} \cdot \|\boldsymbol{k}^{1:L,1} - \boldsymbol{k}^{1:L,2}\|,
\end{aligned}$$

where the first inequality is due to triangular inequality; the second inequality uses the fact that $a_i \le K_{\max}$ and $\mathbb{1}_{\{\boldsymbol{k}^{1,\ell} \neq \boldsymbol{k}^{2,\ell}\}} \le \|\boldsymbol{k}^{1,\ell} - \boldsymbol{k}^{2,\ell}\|$; the third inequality uses triangular inequality, the fact that the total rate at a configuration $\boldsymbol{k}$ is bounded by $\gamma_{\max}$ and the property of the L^1 norm $\|\cdot\|$. Therefore, $f \in \mathrm{Lip}(1)$. The Lipschitz continuity of f allows us to invoke Lemma 2 and get

$$\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^{\ell},(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right|\right] - \mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\overline{\boldsymbol{K}}^{\ell},(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right|\right] \le 2\gamma_{\max}K_{\max} \cdot O\left(\sqrt{r}\right).$$

Therefore,

$$\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^{\ell},(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right|\right] \le \mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\overline{\boldsymbol{K}}^{\ell},(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right|\right] + O\left(\sqrt{r}\right). \tag{45}$$

Observe that under a Markovian policy, the request rate of type i jobs can be written as $\overline{\lambda}_i = \mathbb{E}[\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\overline{\boldsymbol{K}}^{\ell},(\boldsymbol{k}',\boldsymbol{a})} a_i]$, so we have

$$\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\overline{\boldsymbol{K}}^{\ell},(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right] = \overline{\lambda}_i \cdot \lceil \overline{N} \rceil - \lambda_i r = O(1). \tag{46}$$

Moreover, because $\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\overline{\boldsymbol{K}}^{\ell},(\boldsymbol{k}',\boldsymbol{a})} a_i$ are i.i.d. for $\ell = 1, \ldots, L$, we have

$$\begin{aligned}
\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\overline{\boldsymbol{K}}^{\ell},(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right|\right] &\le \sqrt{\mathbb{E}\left[\left(\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\overline{\boldsymbol{K}}^{\ell},(\boldsymbol{k}',\boldsymbol{a})} a_i - \lambda_i r\right)^2\right]} + O(1) \\
&= \sqrt{\sum_{\ell=1}^{L} \mathrm{Var}\left(\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\overline{\boldsymbol{K}}^{\ell},(\boldsymbol{k}',\boldsymbol{a})} a_i\right)} + O(1) \\
&= O\left(\sqrt{r}\right).
\end{aligned}$$

Therefore, by combining the arguments above, we get

$$\begin{aligned}\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}-\lambda_i r\right|\right] &\le \mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i-\lambda_i r\right|\right]+O\left(\sqrt{r}\right)\\ &\le \mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i-\lambda_i r\right|\right]+O\left(\sqrt{r}\right)\\ &\le O\left(\sqrt{r}\right), \end{aligned} \tag{47}$$

which proves (43). This implies that the term in (40) is also in $O\left(\sqrt{r}\right)$.

Combining the bounds on the terms in (40) and (41), we get

$$\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}dv_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right]=O\left(\sqrt{r}\right).$$

Finally, we bound $\mathbb{E}\left[\lambda_i r\cdot dy_i\right]$. We rearrange the terms in (38) and get

$$\begin{aligned}\mathbb{E}\left[\lambda_i r\cdot dy_i\right] &= \mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}(-a_i+dv_i)\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}+\lambda_i r\right]\\ &= \mathbb{E}\left[-\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}+\lambda_i r\right]+O\left(\sqrt{r}\right).\end{aligned}$$

By (43), we have $\mathbb{E}\left[-\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}+\lambda_i r\right]=O\left(\sqrt{r}\right)$. Therefore,

$$\mathbb{E}\left[\lambda_i r\cdot dy_i\right]=O\left(\sqrt{r}\right).$$

We invoke the equations (36) and (37) that we get at the beginning of the proof, and conclude that

$$\mathbb{E}[V_i]\le t_{\max}\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}dv_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right]=O\left(\sqrt{r}\right).$$

$$\mathbb{E}[Y_i]\le t_{\max}\mathbb{E}\left[\lambda_i r\cdot dy_i\right]=O\left(\sqrt{r}\right).$$

This finishes the proof. □

B.2.2 ***Deriving the equality in*** (39). We show the calculation detail of deriving the following equality.

$$\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}dv_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right] \tag{39}$$

$$=\frac{1}{\eta_{\max}}\cdot\mathbb{E}\left[\left(\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}-\lambda_i r\right)\cdot Z_i\right] \tag{40}$$

$$+\frac{1}{2\eta_{\max}}\cdot\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}\left(a_i^2-(dv_i)^2\right)\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}+\lambda_i r\cdot(1-(dy_i)^2)\right]. \tag{41}$$

The equality is obtained by considering the drift of the function $g(z_i) = z_i^2$, which is zero in steady state. Recall that the drift of $g(z_i)$ is given by

$$\begin{aligned}\widehat{G}g(z_i, \boldsymbol{u}^{1:L}) = &\sum_{\ell=1}^{L} \sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(g(z_i + a_i - dv_i) - g(z_i)\right) \mathbb{1}_{\{\boldsymbol{\eta}^\ell = \boldsymbol{0}\}} \\ &+ \lambda_i r \left(g(z_i - 1 + dy_i) - g(z_i)\right). \end{aligned} \tag{35}$$

We will first calculate $g(z_i + a_i - dv_i) - g(z_i)$, then $g(z_i - 1 + dy_i) - g(z_i)$, and finally plug them into the (35).

The calculation of $g(z_i + a_i - dv_i) - g(z_i)$ utilizes the following property of dv_i:

$$(z_i + a_i - dv_i) \cdot dv_i = \eta_{\max} \cdot dv_i. \tag{48}$$

This property follows from the definition $dv_i = (z_i + a_i - \eta_{\max})^+$. Intuitively, this is because dv_i is the "force" that pushes z_i back when it hits the boundary at $\eta_{\max}$. Using the property, we have

$$\begin{aligned} &g(z_i + a_i - dv_i) - g(z_i) \\ &= (z_i + a_i - dv_i)^2 - z_i^2 \\ &= (z_i + a_i - dv_i)^2 - (z_i + a_i - dv_i - a_i + dv_i)^2 \\ &= (z_i + a_i - dv_i)^2 - \left((z_i + a_i - dv_i)^2 + 2(-a_i + dv_i) \cdot (z_i + a_i - dv_i) + (-a_i + dv_i)^2\right) \\ &= 2(a_i - dv_i) \cdot (z_i + a_i - dv_i) - (-a_i + dv_i)^2 \\ &= 2a_i \cdot (z_i + a_i - dv_i) - (-a_i + dv_i)^2 - 2dv_i \cdot \eta_{\max} \\ &= 2a_i \cdot z_i + a_i^2 - (dv_i)^2 - 2dv_i \cdot \eta_{\max}. \end{aligned}$$

The second last equality is due to (48), and the rest are all algebraic manipulations.

We carry out a similar calculation for $g(z_i - 1 + dy_i) - g(z_i)$:

$$\begin{aligned} &g(z_i - 1 + dy_i) - g(z_i) \\ &= 2(-1 + dy_i) \cdot z_i + (-1 + dy_i)^2 \\ &= -2z_i + 2z_i \cdot dy_i + 1 - 2dy_i + (dy_i)^2 \\ &= -2z_i + 1 + 2(z_i - 1 + dy_i) \cdot dy_i - (dy_i)^2 \\ &= -2z_i + 1 - (dy_i)^2. \end{aligned}$$

where the last equality is due to the property that

$$(z_i - 1 + dy_i) \cdot dy = 0, \tag{49}$$

and the rest are all algebraic manipulations.

Putting together,

$$\begin{aligned}\widehat{G}g(z_i, \boldsymbol{u}^{1:L}) &= \sum_{\ell=1}^{L} \sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(g(z_i + a_i - dv_i) - g(z_i)\right) \mathbb{1}_{\{\boldsymbol{\eta}^\ell = \boldsymbol{0}\}} \\ &\quad + \lambda_i r \left(g(z_i - 1 + dy_i) - g(z_i)\right) \\ &= \sum_{\ell=1}^{L} \sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(2a_i \cdot z_i + a_i^2 - (dv_i)^2 - 2dv_i \cdot \eta_{\max}\right) \mathbb{1}_{\{\boldsymbol{\eta}^\ell = \boldsymbol{0}\}} \\ &\quad + \lambda_i r \cdot \left(-2z_i + 1 - (dy_i)^2\right). \end{aligned}$$

After recombining the terms, we get

$$\begin{aligned}
&\eta_{\max} \cdot \mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} dv_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right] \\
=\ &\mathbb{E}\left[\left(\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} - \lambda_i r\right)\cdot Z_i\right] \\
&+\frac{1}{2}\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(a_i^2-(dv_i)^2\right)\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} + \lambda_i r\cdot(1-(dy_i)^2)\right]
\end{aligned}$$

This finishes the calculation.

B.3 Proof of Conversion Theorem without Assuming Irreducibility

In this subsection, we prove Theorem 3 without assuming $\boldsymbol{k}^0$-irreducibility of the subroutine $\overline{\sigma}$. Specifically, suppose that we have a Markovian single-server policy $\overline{\sigma}$ and an initial distribution p_j over its recurrent classes S_j for $j = 1, 2 \ldots J$, we will construct an infinite-server policy σ such that (5) and (6) still hold. The basic idea is to decompose a general Markovian single-server policy $\overline{\sigma}$ into multiple $\boldsymbol{k}^0$-irreducible Markovian policies, each induces one recurrent class and preserves stationary distribution and the throughput of $\overline{\sigma}$ on that recurrent class, as stated in Lemma 5 below.

Lemma 5 (Decomposing The Reducible Policy)**.** *Let $\overline{\sigma}$ be a general single-server Markovian policy with recurrent classes S_j for $j = 1, 2, \ldots, J$. Then for each j exists a Markovian policy $\overline{\sigma}^j$ such that*

- *The induced Markov chain is $\boldsymbol{k}^0$-irreducible with the unique recurrent class being S_j;*
- *The stationary distribution is the same as the stationary distribution under $\overline{\sigma}$ starting from a configuration in S_j.*

Proof. For each $j = 1, 2, \ldots, J$, we define the policy $\overline{\sigma}^j$ as follows: when the system has configuration $\boldsymbol{k} \in S_j$, the policy $\overline{\sigma}^j$ makes the same decisions as $\overline{\sigma}$; when $\boldsymbol{k} \notin S_j$ and $\boldsymbol{k} = \boldsymbol{0}$, the policy starts a timer whose duration follows an exponential distribution with rate 1 and immediately adds $\boldsymbol{k}^0$ many jobs of each type when the timer ticks for some arbitrary $\boldsymbol{k}^0 \in S_j$; when $\boldsymbol{k} \notin S_j$ and $\boldsymbol{k} \neq \boldsymbol{0}$, the policy does not request any jobs.

We show that under the new policy $\overline{\sigma}^j$, S_j is also a recurrent class of the induced Markov chain. This is because if the system starts from a configuration $\boldsymbol{k} \in S_j$, then it will stay in S_j since it makes the same decisions and has the same transitions as under the policy $\overline{\sigma}$. Because S_j is a recurrent class under $\overline{\sigma}$, it is still a recurrent class under $\overline{\sigma}^j$.

To show that the Markov chain induced by $\overline{\sigma}^j$ is $\boldsymbol{k}^0$-irreducible, observe that starting from any $\boldsymbol{k} \notin S_j$, the system state will return to S_j. Specifically,

- If the system starts from a configuration $\boldsymbol{k}$ such that $\boldsymbol{k} \notin S_j$ and $\boldsymbol{k} \neq \boldsymbol{0}$, then no new jobs will be requested until either $\boldsymbol{k} \in S_j$ or $\boldsymbol{k} = \boldsymbol{0}$. In the latter case, by the construction of the policy, the system jumps to a configuration in S_j after the next transition.
- If the system starts from a configuration $\boldsymbol{k}$ such that $\boldsymbol{k} \notin S_j$ and $\boldsymbol{k} = \boldsymbol{0}$, the system jumps to a configuration in S_j after the next transition.

The claim that the stationary distribution under $\overline{\sigma}^j$ is the same as the stationary distribution under $\overline{\sigma}$ starting from a configuration in S_j is trivial to show, because when the system initializes from any configuration in S_j, it stays in S_j and the transitions are exactly the same under the two policies. □

The JRS policy with a general Markovian single-server policy as its subroutine $\overline{\sigma}$ is constructed using the $\boldsymbol{k}^0$-irreducible policies $\overline{\sigma}^j$'s obtained from the decomposition of $\overline{\sigma}$ and the probabilities p^j's.

(1) We divide all the servers into J server pools, each with infinitely many servers. Let the j-th server pool run the JRS policy with subroutine $\overline{\sigma}^j$ (defined in Section 4.2) for each $j = 1, \dots J$.
(2) Whenever we see an arrival of type i, we route the job to the j-th infinite-server system with probability $\frac{p^j \overline{\lambda}_i^j}{\sum_{j'} p^{j'} \overline{\lambda}_{i'}^{j'}}$ for each $j = 1, \dots J$.

To analyze the policy σ, let π^j and $\overline{\lambda}_i^j$'s be the stationary distribution and throughput of the policy $\overline{\sigma}^j$ for $j = 1, \dots, J$. By Lemma 5, we have the following relationships:

$$\pi(\boldsymbol{k}) = \textstyle\sum_{j=1}^J p^j \pi^j(\boldsymbol{k}), \quad \forall \boldsymbol{k} \in \mathcal{K}, \tag{50}$$

$$\overline{\lambda}_i = \textstyle\sum_{j=1}^J p^j \overline{\lambda}_i^j, \quad \forall i \in \mathcal{I}. \tag{51}$$

Based on the above relationships, we can prove the general version of the Conversion Theorem that does not require $\boldsymbol{k}^0$-irreducibility.

Proof of Theorem 3. For each $j = 1, \dots, J$, Lemma 5 implies that the policy $\overline{\sigma}^j$ and stationary distribution π^j form a feasible solution to the single-server problem $\overline{\mathcal{P}}\left(\left(p^j \overline{\lambda}_i^j \overline{N}\right)_{i \in \mathcal{I}}, \epsilon\right)$, and the corresponding objective value is $p^j \overline{N}$. Consider the infinite-server system with arrival rates $\left(p^j \overline{\lambda}_i^j \overline{N}\right)_{i \in \mathcal{I}}$ and budget ϵ. As we have proved Theorem 3 for the JRS policy with $\boldsymbol{k}^0$-irreducible subroutine $\overline{\sigma}^j$, it follows that

$$\left| \textstyle\sum_{\boldsymbol{k} \neq \boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}^j] - \left\lceil p^j \overline{N} \right\rceil \cdot (1 - \pi^j(\boldsymbol{0})) \right| = O\left(\sqrt{r}\right), \tag{52}$$

$$\left| \textstyle\sum_{\boldsymbol{k} \neq \boldsymbol{0}} h(\boldsymbol{k}) \mathbb{E}[X_{\boldsymbol{k}}^j] - \left\lceil p^j \overline{N} \right\rceil \cdot \sum_{\boldsymbol{k} \neq \boldsymbol{0}} h(\boldsymbol{k}) \pi^j(\boldsymbol{k}) \right| = O\left(\sqrt{r}\right), \tag{53}$$

where $X_{\boldsymbol{k}}^j$ is the random variable representing the steady-state number of servers in configuration $\boldsymbol{k}$ in the infinite-server system under the JRS policy with subroutine $\overline{\sigma}^j$.

By the construction of σ, the arrival rate to the j-th server pool is equal to $\frac{p^j \overline{\lambda}_i^j}{\sum_{j'} p^{j'} \overline{\lambda}_{i'}^{j'}} \cdot \lambda_i r = \frac{p^j \overline{\lambda}_i^j}{\overline{\lambda}_i} \cdot \lambda_i r = p^j \overline{\lambda}_i^j \overline{N}$, where the first equality is due to (51), and the second equality is due to the condition that $\lambda_i r = \overline{\lambda}_i^j \cdot L$. Therefore, (52) and (53) hold, and we have

$$\begin{aligned}
\left| \sum_{\boldsymbol{k} \neq \boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}] - \overline{N} \cdot (1 - \pi(\boldsymbol{0})) \right| &= \left| \sum_{j=1}^J \sum_{\boldsymbol{k} \neq \boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}^j] - \sum_{j=1}^J p^j \overline{N} \cdot (1 - \pi^j(\boldsymbol{0})) \right| \\
&\leq \sum_{j=1}^J \left| \sum_{\boldsymbol{k} \neq \boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}^j] - \left\lceil p^j \overline{N} \right\rceil \cdot (1 - \pi^j(\boldsymbol{0})) \right| + O(1) \\
&= O(\sqrt{r}).
\end{aligned}$$

Here we use (52) and the relationship between $\pi(\boldsymbol{k})$ and $\pi^j(\boldsymbol{k})$. Similarly,

$$\left| \sum_{\boldsymbol{k} \neq \boldsymbol{0}} h(\boldsymbol{k}) \mathbb{E}[X_{\boldsymbol{k}}] - \overline{N} \cdot \sum_{j=1}^J \sum_{\boldsymbol{k} \neq \boldsymbol{0}} h(\boldsymbol{k}) \pi(\boldsymbol{k}) \right| = \left| \sum_{j=1}^J \sum_{\boldsymbol{k} \neq \boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}^j] - \sum_{j=1}^J p^j \overline{N} \cdot \sum_{\boldsymbol{k} \neq \boldsymbol{0}} h(\boldsymbol{k}) \pi^j(\boldsymbol{k}) \right|$$

$$= \sum_{j=1}^{J} \left| \sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}[X_{\boldsymbol{k}}^j] - \left\lceil p^j \overline{N} \right\rceil \cdot \sum_{\boldsymbol{k}\neq\boldsymbol{0}} h(\boldsymbol{k})\pi^j(\boldsymbol{k}) \right| + O(1)$$
$$= O(\sqrt{r}).$$

After re-indexing the servers, we get (5) and (6). The bounds on $N(\sigma)$ and $C(\sigma)$, (7) and (8), follow from (5) and (6). They can be verified in the same way as that in the proof for the irreducible case, so we omit the argument here. □

C SOLVING THE SINGLE-SERVER PROBLEM

In this section, we show the equivalence of the single-server problem in (3) and a linear program (LP) as stated in Theorem 4. The equivalence needs to be proved in two directions. In Appendix C.1, we first derive the linear program (61) as a relaxation of the single-server problem (3) so that the optimal value of the LP is a lower bound to the optimal value of (3). Then in Appendix C.2, we will construct a Markovian single-server policy that achieves the optimal value of the LP, which implies the optimality of the policy.

C.1 Lower Bound via LP Relaxation

In this subsection we derive an LP relaxation of the optimization problem (3), restated below:

$$\begin{aligned} \underset{\overline{N},\ \overline{\sigma},\ \pi}{\text{minimize}} \quad & \overline{N} \\ \text{subject to} \quad & \mathbb{E}\left[h\big(\overline{\boldsymbol{K}}(\infty)\big)\middle|\overline{\boldsymbol{K}}(\infty)\neq\boldsymbol{0}\right] \le \epsilon, \\ & \overline{N}\cdot\overline{\lambda}_i = \lambda_i r, \quad \forall i\in\mathcal{I}. \end{aligned} \tag{3 revisited}$$

Here we allow $\overline{\sigma}$ to be non-Markovian, i.e., it can make decisions based on the history, but we still require its performance metrics used in the objective and the constraints of (3) to have well-defined steady-state distributions.

Observe that both $\overline{\boldsymbol{K}}(\infty)$ and $\overline{\lambda}_i$ depend on the stationary distribution π, but the constraints in terms of π are implicit. To derive an LP relaxation, we give an explicit characterization of the constraints that must be satisfied by the stationary distribution π induced by any feasible policy $\overline{\sigma}$.

To do this, we derive a version of the stationary equation in terms of a quantity called *transition frequency*. The transition frequency of type i jobs is a function $u_i\colon \mathcal{K}\to\mathbb{R}$ describing the steady-state frequency of requesting a type i job when the system has configuration $\boldsymbol{k}$. To rigorously define transition frequency, we first introduce a concept called *nominal transition*.

Definition 1 (Nominal Transition)**.** Consider a single-server system under any policy. When the configuration $\overline{\boldsymbol{K}}(t)$ transitions from $\boldsymbol{k}$ to $\boldsymbol{k}'+\boldsymbol{a}$ for some $\boldsymbol{k},\boldsymbol{k}'\in\mathcal{K}$ with $\boldsymbol{a}=(a_i)_{i\in\mathcal{I}}$ new jobs added into service, we decompose the transition by adding intermediate configurations as illustrated below, where $\boldsymbol{k}$ first goes to $\boldsymbol{k}'$ if $\boldsymbol{k}'\neq\boldsymbol{k}$, then add jobs of each type one by one.

$$\boldsymbol{k}\to\boldsymbol{k}'\to(\boldsymbol{k}'+\boldsymbol{e}_{i_1})\to\cdots\to(\boldsymbol{k}'+a_{i_1}\boldsymbol{e}_{i_1})\to\cdots\to(\boldsymbol{k}'+a_{i_1}\boldsymbol{e}_{i_1}+\cdots+a_{i_{|\mathcal{I}|}}\boldsymbol{e}_{i_{|\mathcal{I}|}}),$$

where $(i_1,i_2,\ldots,i_{|\mathcal{I}|})$ is a fixed ordering of the set of phases $\mathcal{I}$. We call each short transition in the diagram a *nominal transition*.

For $\boldsymbol{k}^1,\boldsymbol{k}^2\in\mathcal{K}$, we denote $F(\boldsymbol{k}^1,\boldsymbol{k}^2,t)$ as the cumulative number of nominal transitions from $\boldsymbol{k}^1$ to $\boldsymbol{k}^2$ during the time interval $[0,t]$, which is a random variable with a distribution depending on the single-server policy and initial distribution of configurations.

Note that for any $i \in \mathcal{I}$ and $\boldsymbol{k} \in \mathcal{K}$ s.t. $k_i \geq 1$, $F(\boldsymbol{k}, \boldsymbol{k} - \boldsymbol{e}_i, t)$ counts the number of times that a type i job departs when being in configuration $\boldsymbol{k}$. As a result,

$$F(\boldsymbol{k}, \boldsymbol{k} - \boldsymbol{e}_i, t) = \mathcal{N}\left(\int_0^t k_i \mu_{i\perp} \mathbb{1}_{\{K(s)=\boldsymbol{k}\}} ds\right),$$

where $\mathcal{N}(t)$ denotes a unit rate Poisson process. If we take expectation, divide both sides by t, and let $t \to \infty$, we have

$$\lim_{t\to\infty} \frac{1}{t}\mathbb{E}[F(\boldsymbol{k}, \boldsymbol{k} - \boldsymbol{e}_i, t)] = k_i \mu_{i\perp} \cdot \lim_{t\to\infty} \frac{1}{t} \int_0^t \mathbb{P}(\boldsymbol{K}(s) = \boldsymbol{k}) ds = k_i \mu_{i\perp} \pi(\boldsymbol{k}). \tag{54}$$

Similarly, $F(\boldsymbol{k}, \boldsymbol{k} - \boldsymbol{e}_i + \boldsymbol{e}_{i'}, t)$ counts the number of times a job in phase i transitions to phase i' when being in configuration $\boldsymbol{k}$ for any $i, i' \in \mathcal{I}$, $\boldsymbol{k} \in \mathcal{K}$ s.t. $i' \neq i$ and $k_i \geq 1$, so

$$\lim_{t\to\infty} \frac{1}{t}\mathbb{E}[F(\boldsymbol{k}, \boldsymbol{k} - \boldsymbol{e}_i + \boldsymbol{e}_{i'}, t)] = k_i \mu_{ii'} \pi(\boldsymbol{k}). \tag{55}$$

We define transition frequency as follows.

Definition 2 (Transition Frequency)**.** Transition frequency of type i jobs at state $\boldsymbol{k}$ is the long-run average number of nominal transitions from configuration $\boldsymbol{k}$ to $\boldsymbol{k} + \boldsymbol{e}_i$ per unit time,

$$u_i(\boldsymbol{k}) \triangleq \lim_{t\to\infty} \frac{1}{t}\mathbb{E}[F(\boldsymbol{k}, \boldsymbol{k} + \boldsymbol{e}_i, t)]. \tag{56}$$

The transition frequencies allow us to derive the following version of the stationary equation.

Lemma 6 (Stationary Equation)**.** *Under any policy, the stationary distribution π and the transition frequency u_i satisfy the following equation:*

$$\begin{aligned} &\sum_i u_i(\boldsymbol{k} - \boldsymbol{e}_i)\mathbb{1}_{\{k_i \geq 1\}} + \sum_i (k_i + 1)\mu_{i\perp}\pi(\boldsymbol{k} + \boldsymbol{e}_i) \\ &+ \sum_{i,i' :\, i \neq i'} (k_i + 1)\mu_{ii'}\pi(\boldsymbol{k} + \boldsymbol{e}_i - \boldsymbol{e}_{i'})\mathbb{1}_{\{k_{i'} \geq 1\}} \\ &= \sum_i u_i(\boldsymbol{k}) + \Big(\sum_i k_i \mu_{i\perp} + \sum_{i,i' :\, i \neq i'} k_i \mu_{ii'}\Big)\pi(\boldsymbol{k}) \end{aligned} \tag{57}$$

for any state $\boldsymbol{k} \in \mathcal{K}$, and $\sum_i, \sum_{i,i'}$ are shorthand for $\sum_{i\in\mathcal{I}}, \sum_{i,i'\in\mathcal{I}}$.

Proof. For each configuration $\boldsymbol{k} \in \mathcal{K}$, if we look at the difference between the number of nominal transitions into configuration $\boldsymbol{k}$ and that out of configuration $\boldsymbol{k}$ by time t, we have the following equation,

$$\begin{aligned} &\mathbb{1}_{\{\overline{\boldsymbol{K}}(t)=\boldsymbol{k}\}} - \mathbb{1}_{\{\overline{\boldsymbol{K}}(0)=\boldsymbol{k}\}} \\ &= \sum_i F(\boldsymbol{k} - \boldsymbol{e}_i, \boldsymbol{k}, t)\mathbb{1}_{\{k_i \geq 1\}} + \sum_i F(\boldsymbol{k} + \boldsymbol{e}_i, \boldsymbol{k}, t) + \sum_{i,i' :\, i \neq i'} F(\boldsymbol{k} + \boldsymbol{e}_i - \boldsymbol{e}_{i'}, \boldsymbol{k}, t)\mathbb{1}_{\{k_{i'} \geq 1\}} \\ &\quad - \sum_i F(\boldsymbol{k}, \boldsymbol{k} + \boldsymbol{e}_i, t) - \sum_i F(\boldsymbol{k}, \boldsymbol{k} - \boldsymbol{e}_i, t)\mathbb{1}_{\{k_i \geq 1\}} - \sum_{i,i' :\, i \neq i'} F(\boldsymbol{k}, \boldsymbol{k} - \boldsymbol{e}_i + \boldsymbol{e}_{i'}, t)\mathbb{1}_{\{k_i \geq 1\}} \end{aligned} \tag{58}$$

By Definition 2 and (54)-(55), if we divide both sides of (58) by t and let $t \to \infty$, we get the stationary equation in (57). □

Since the stationary equation in (57) is linear in $u_i(\boldsymbol{k})$ and $\pi(\boldsymbol{k})$, we can write it in matrix form:

$$A\boldsymbol{\pi} + \textstyle\sum_{i\in\mathcal{I}} B_i \boldsymbol{u}_i = 0, \tag{59}$$

where $\boldsymbol{\pi}$ and $\boldsymbol{u}_i$ are column vectors representing $\pi(\cdot)$, $u_i(\cdot)$, and A, B_i are matrices that make (59) equivalent to (57). Therefore, the following three conditions are necessary for any tuple $(\boldsymbol{\pi}, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$ to be a possible pair of stationary distribution and transition frequencies for a Markovian policy.

$$\begin{aligned} A\boldsymbol{\pi} + \textstyle\sum_{i\in\mathcal{I}} B_i \boldsymbol{u}_i &= 0 \\ \textstyle\sum_{\boldsymbol{k}} \pi(\boldsymbol{k}) &= 1 \\ \boldsymbol{\pi}, \boldsymbol{u}_i &\geq 0, \quad \forall i \in \mathcal{I} \end{aligned} \tag{60}$$

Based on the characterization of stationary $\boldsymbol{\pi}$ and $\boldsymbol{u}_i$'s in (60), we can now formulate a linear program $\overline{\mathcal{LP}}((\lambda_i)_{i\in\mathcal{I}}, \epsilon)$. The linear program has decision variables $\Phi \in \mathbb{R}$, $\boldsymbol{\pi} \in \mathbb{R}^{\mathcal{K}}$, and $\boldsymbol{u}_i \in \mathbb{R}^{\mathcal{K}}$ for $i \in \mathcal{I}$, where Φ is a factor that scales the throughput of each type of jobs in the direction of $(\lambda_i)_{i\in\mathcal{I}}$.

$$\begin{aligned} \underset{\Phi,\ \boldsymbol{\pi},\ (\boldsymbol{u}_i)_{i\in\mathcal{I}}}{\text{maximize}} \quad & \Phi \\ \text{subject to} \quad & \boldsymbol{h}^T\boldsymbol{\pi} \leq \epsilon(1 - \pi(\mathbf{0})) \\ & \mathbf{1}_o^T \boldsymbol{u}_i = \Phi \cdot \lambda_i \quad \forall i \\ & A\boldsymbol{\pi} + \sum\nolimits_{i\in\mathcal{I}} B_i \boldsymbol{u}_i = \mathbf{0} \\ & \mathbf{1}^T\boldsymbol{\pi} = 1 \\ & \boldsymbol{\pi} \geq 0, \boldsymbol{u}_i \geq 0 \quad \forall i \in \mathcal{I} \end{aligned} \tag{61}$$

where $\boldsymbol{h}$ is the vector form of the cost rate function h; $\mathbf{1}_o^T$ is a $|\mathcal{K}|$-dimensional vector with one in all entries except those with $\sum_{i\in\mathcal{I}} k_i = K_{\max}$. In addition to the last three constraints on stationarity, the first constraint of $\overline{\mathcal{LP}}((\lambda_i)_{i\in\mathcal{I}}, \epsilon)$ is the resource contention constraint; the second constraint is because the transition frequency from $\boldsymbol{k}$ to $\boldsymbol{k} + \boldsymbol{e}_i$ is equal to the throughput of type i jobs.

By the construction of $\overline{\mathcal{LP}}((\lambda_i)_{i\in\mathcal{I}}, \epsilon)$, any feasible solution of $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ can be converted to a feasible solution of $\overline{\mathcal{LP}}((\lambda_i)_{i\in\mathcal{I}}, \epsilon)$. Let $\overline{N}^*$ be the optimal value of (3) and Φ^* be the optimal value of (61). Then we have

$$\overline{N}^* \geq \frac{r}{\Phi^*}. \tag{62}$$

C.2 Policy Construction

In this subsection, we describe a procedure that allows us to construct a policy that achieves the lower bound given by the LP relaxation in (61). Specifically, given a feasible solution $(\boldsymbol{\pi}, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$ to (61), we define an *LP-based policy* that requests jobs as follows:

- Case 1: When the system enters a configuration $\boldsymbol{k}$ with $\pi(\boldsymbol{k}) \neq 0$, for each $i \in \mathcal{I}$, the policy starts a timer whose duration follows an exponential distribution with rate $\frac{u_i(\boldsymbol{k})}{\pi(\boldsymbol{k})}$. The policy requests a type i job when the i-th timer ticks. When the configuration changes, all timers are canceled.
- Case 2: When the system enters a configuration $\boldsymbol{k}$ with $\pi(\boldsymbol{k}) = 0$ and $\sum_{i'} u_{i'}(\boldsymbol{k}) \neq 0$, the policy immediately requests a type i job with probability $\frac{u_i(\boldsymbol{k})}{\sum_{i'} u_{i'}(\boldsymbol{k})}$.
- Case 3: When the system enters a configuration $\boldsymbol{k}$ with $\pi(\boldsymbol{k}) = 0$ and $\sum_{i'} u_{i'}(\boldsymbol{k}) = 0$, the policy does not request any jobs.

We denote the LP-based policy based on the solution $(\boldsymbol{\pi}, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$ as $\overline{\sigma}(\boldsymbol{\pi}, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$.

Remark 2. Note that the definition of the LP-based policy here is stated from a view different from the view in Section 4.2: here each request only adds one job to the server, and one request can happen immediately after another; while in Section 4.2 each request can add multiple jobs to the server, and there is only one request happening at a time. We refer to the view here as the *impulsive*

view, because multiple requests happening at the same time can be thought of as having an infinite request rate. In contrast, we call the view in Section 4.2 the *non-impulsive view*.

The LP-based policy can be alternatively described using the non-impulsive view if we see multiple requests happening at the same time as one request that adds multiple jobs to the server. More specifically, each reactive request of the LP-based policy is initiated by an internal transition or departure and consists of one or multiple requests of Case 2; each proactive request of the LP-based policy consists of one request in Case 1 and possibly several requests in Case 2.

The following lemma characterizes the steady-state behavior of a single-server system under the LP-based policy.

Lemma 7 (Properties of LP-based Policies)**.** *Consider a single-server system under the LP-based policy $\overline{\sigma}(\boldsymbol{\pi}, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$, where $(\boldsymbol{\pi}, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$ is a feasible solution to (61). We have that $\boldsymbol{\pi}$ is a stationary distribution under policy $\overline{\sigma}$, and $(\boldsymbol{u}_i)_{i\in\mathcal{I}}$ are the transition frequencies corresponding to $\overline{\sigma}$.*

The proof of Lemma 7 is based on (58), following the same argument as the proof of Lemma 6, as well as an induction argument.

Proof. Let $(\boldsymbol{\pi}, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$ be a feasible solution to the LP in (61). To show that $\boldsymbol{\pi}$ and $(\boldsymbol{u}_i)_{i\in\mathcal{I}}$ are also the actual stationary distribution and transition frequencies, it suffices to show that if the initial distribution follows $\boldsymbol{\pi}$, i.e.

$$\mathbb{P}\big(\overline{\boldsymbol{K}}(0) = \boldsymbol{k}\big) = \pi(\boldsymbol{k})$$

then under the policy $\overline{\sigma}(\boldsymbol{\pi}, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$, we have

$$\lim_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{P}\big(\overline{\boldsymbol{K}}(t) = \boldsymbol{k}\big)dt = \pi(\boldsymbol{k}), \tag{63}$$

$$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}[F(\boldsymbol{k}, \boldsymbol{k}+\boldsymbol{e}_i, T)] = u_i(\boldsymbol{k}), \tag{64}$$

where F is the cumulative number of nominal transitions under the policy $\overline{\sigma}(\boldsymbol{\pi}, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$ and the initial distribution $\boldsymbol{\pi}$.

Our proof is based on the following equation:

$$\begin{aligned}
&\frac{d}{dt}\mathbb{P}(\overline{\boldsymbol{K}}(t) = \boldsymbol{k})\Big|_{t=0} \\
&= \sum_i \frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k}-\boldsymbol{e}_i, \boldsymbol{k}, t)\right]\Big|_{t=0} - \sum_i \frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k}, \boldsymbol{k}+\boldsymbol{e}_i, t)\right]\Big|_{t=0} \\
&\quad + \sum_i (k_i+1)\mu_{i\perp}\pi(\boldsymbol{k}+\boldsymbol{e}_i) + \sum_{i,i'} (k_i+1)\mu_{ii'}\pi(\boldsymbol{k}+\boldsymbol{e}_i-\boldsymbol{e}_{i'})\mathbb{1}_{\{k_{i'}\geq 1\}} \\
&\quad - \sum_i k_i\mu_{i\perp}\pi(\boldsymbol{k}) - \sum_{i,i'} k_i\mu_{ii'}\pi(\boldsymbol{k}),
\end{aligned} \tag{65}$$

The equation is a straightforward consequence of (58), following the same argument as the proof of Lemma 6.

We prove the following two equations by induction on $\sum_{i\in\mathcal{I}} k_i$.

$$\lim_{t\to 0} \frac{1}{t}\mathbb{E}[F(\boldsymbol{k}, \boldsymbol{k}+\boldsymbol{e}_i, t)] = u_i(\boldsymbol{k}), \tag{66}$$

$$\frac{d}{dt}\mathbb{P}(\overline{\boldsymbol{K}}(t) = \boldsymbol{k})|_{t=0} = 0. \tag{67}$$

We first consider the base case when $\sum_{i\in\mathcal{I}} k_i = 0$. In this case, $\boldsymbol{k} = \boldsymbol{0}$ and we have

$$\frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k}-\boldsymbol{e}_i,\boldsymbol{k},t)\right]\Big|_{t=0} = u_i(\boldsymbol{k}-\boldsymbol{e}_i)\mathbb{1}_{\{k_i\geq 1\}} = 0, \tag{68}$$

for all i. This reduces (65) to

$$\begin{aligned}
&\frac{d}{dt}\mathbb{P}(\overline{\boldsymbol{K}}(t)=\boldsymbol{k})\Big|_{t=0}\\
=&\sum_i u(\boldsymbol{k}-\boldsymbol{e}_i)\mathbb{1}_{\{k_i\geq 1\}} - \sum_i \frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k},\boldsymbol{k}+\boldsymbol{e}_i,t)\right]\Big|_{t=0}\\
&+\sum_i (k_i+1)\mu_{i\perp}\pi(\boldsymbol{k}+\boldsymbol{e}_i) + \sum_{i,i'}(k_i+1)\mu_{i,i'}\pi(\boldsymbol{k}+\boldsymbol{e}_i-\boldsymbol{e}_{i'})\mathbb{1}_{\{k_{i'}\geq 1\}}\\
&-\sum_i k_i\mu_{i\perp}\pi(\boldsymbol{k}) - \sum_{i,i'} k_i\mu_{i,i'}\pi(\boldsymbol{k}),
\end{aligned} \tag{69}$$

Now we discuss based on whether $\pi(\boldsymbol{k}) = 0$. If $\pi(\boldsymbol{k}) \neq 0$, by the definition of our policy, for all i,

$$\frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k},\boldsymbol{k}+\boldsymbol{e}_i,t)\right]\Big|_{t=0} = \frac{u_i(\boldsymbol{k})}{\pi(\boldsymbol{k})}\cdot\mathbb{P}(\overline{\boldsymbol{K}}(0)=\boldsymbol{k}) = u_i(\boldsymbol{k}), \tag{70}$$

which is (66). Combining the above equation and the stationary equation (57) satisfied by $(\boldsymbol{\pi}, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$, we conclude that the RHS of (69) is zero, i.e.

$$\frac{d}{dt}\mathbb{P}(\overline{\boldsymbol{K}}(t)=\boldsymbol{k})|_{t=0} = 0,$$

which is (67). For the case when $\pi(\boldsymbol{k}) = 0$ and $\sum_i u_i(\boldsymbol{k}) \neq 0$, because the system immediately leave the configuration $\boldsymbol{k}$ after reaching it through a nominal transition,

$$\frac{d}{dt}\mathbb{P}(\overline{\boldsymbol{K}}(t)=\boldsymbol{k})|_{t=0} = 0,$$

i.e., the LHS of (69) is 0. Again we compare (69) against the stationary equation (57) and get

$$\sum_i \frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k},\boldsymbol{k}+\boldsymbol{e}_i,t)\right]\Big|_{t=0} = \sum_i u_i(\boldsymbol{k}).$$

By the definition of our policy, we have

$$\frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k},\boldsymbol{k}+\boldsymbol{e}_i,t)\right]\Big|_{t=0} = \frac{u_i(\boldsymbol{k})}{\sum_{i'}u_{i'}(\boldsymbol{k})}\cdot\sum_{i'}\frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k},\boldsymbol{k}+\boldsymbol{e}_{i'},t)\right]\Big|_{t=0} = u_i(\boldsymbol{k}). \tag{71}$$

which is (66). For the case when $\pi(\boldsymbol{k}) = 0$ and $\sum_i u_i(\boldsymbol{k}) = 0$, (57) implies that $u_i(\boldsymbol{k}-\boldsymbol{e}_i) = 0$, $\pi(\boldsymbol{k}+\boldsymbol{e}_i) = 0$, $\pi(\boldsymbol{k}+\boldsymbol{e}_i-\boldsymbol{e}_{i'}) = 0$ for any i. Then (69) is further reduced to

$$\frac{d}{dt}\mathbb{P}(\overline{\boldsymbol{K}}(t)=\boldsymbol{k})\Big|_{t=0} = -\sum_i \frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k},\boldsymbol{k}+\boldsymbol{e}_i,t)\right]\Big|_{t=0}.$$

Because $\mathbb{P}(\overline{\boldsymbol{K}}(t)=\boldsymbol{k}) \geq 0$ and $\mathbb{P}(\overline{\boldsymbol{K}}(0)=\boldsymbol{k}) = 0$, the LHS of the above expression is non-negative. However, the RHS of the above expression is non-positive. Therefore, both sides are equal to zero, thus we have $\frac{d}{dt}\mathbb{P}(\overline{\boldsymbol{K}}(t)=\boldsymbol{k})\Big|_{t=0} = 0$ and $\frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k},\boldsymbol{k}+\boldsymbol{e}_i,t)\right]\Big|_{t=0} = u_i(\boldsymbol{k})$.

Having proved the base case, we do the induction step. Suppose we have proved (66) and (67) for all $\boldsymbol{k}$ such that $\sum_{i\in\mathcal{I}} k_i \leq m-1$ for some integer $m \geq 1$. We consider $\boldsymbol{k}$ with $\sum_{i\in\mathcal{I}} k_i = m$. By the induction hypothesis,

$$\frac{d}{dt}\mathbb{E}\left[F(\boldsymbol{k}-\boldsymbol{e}_i,\boldsymbol{k},t)\right]\Big|_{t=0} = u_i(\boldsymbol{k}-\boldsymbol{e}_i)\mathbb{1}_{\{k_i\geq 1\}}. \tag{72}$$

Then we repeat the arguments after (68) of the base case verbatim. By induction, we have proved (66) and (67).

Therefore, given the policy and initial distribution, the distribution of $\overline{\boldsymbol{K}}(t)$ is stationary, i.e., we always have $\mathbb{P}(\overline{\boldsymbol{K}}(t) = \boldsymbol{k}) = \pi(\boldsymbol{k})$ for all $\boldsymbol{k} \in \mathcal{K}$. As a result, an analogue of (66) holds for all $t \geq 0$: $\mathbb{E}[F(\boldsymbol{k}, \boldsymbol{k} + \boldsymbol{e}_i, t)]$ is differentiable with respect to t and

$$\frac{d}{dt}\mathbb{E}[F(\boldsymbol{k}, \boldsymbol{k} + \boldsymbol{e}_i, t)] = u_i(\boldsymbol{k}),$$

for all $\boldsymbol{k} \in \mathcal{K}$ and all $i \in \mathcal{I}$. Therefore,

$$\begin{aligned}\lim_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{P}(\overline{\boldsymbol{K}}(t) = \boldsymbol{k})dt &= \lim_{T\to\infty} \frac{1}{T} \cdot \pi(\boldsymbol{k}) \cdot T = \pi(\boldsymbol{k}),\\ \lim_{T\to\infty} \frac{1}{T}\mathbb{E}[F(\boldsymbol{k}, \boldsymbol{k} + \boldsymbol{e}_i, T)] &= \lim_{T\to\infty} \frac{1}{T}\int_0^T \frac{d}{dt}\mathbb{E}[F(\boldsymbol{k}, \boldsymbol{k} + \boldsymbol{e}_i, t)]dt\\ &= \lim_{T\to\infty} \frac{1}{T} \cdot u_i(\boldsymbol{k}) \cdot T = u_i(\boldsymbol{k}).\end{aligned}$$

This completes the proof. □

C.3 Proof of Theorem 4

Theorem 4 (Optimality of Single-OPT)**.** *Given an optimal solution $(\Phi^*, \boldsymbol{\pi}^*, (\boldsymbol{u}_i)_{i\in\mathcal{I}})$ to the linear program $\overline{\mathcal{LP}}((\lambda_i)_{i\in\mathcal{I}}, \epsilon)$, we can solve the single-server problem $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ in (3) to achieve an optimal value r/Φ^*, using the optimal policy $\overline{\sigma}(\boldsymbol{\pi}^*, (\boldsymbol{u}_i)^*_{i\in\mathcal{I}})$ and the optimal stationary distribution $\boldsymbol{\pi}^*$. Moreover, the optimal policy $\overline{\sigma}(\boldsymbol{\pi}^*, (\boldsymbol{u}_i)^*_{i\in\mathcal{I}})$ is a Markovian policy.*

Proof. By Lemma 7, under the policy $\overline{\sigma}(\boldsymbol{\pi}^*, (\boldsymbol{u}_i^*)_{i\in\mathcal{I}})$, $\boldsymbol{\pi}^*$ is a stationary distribution, and $(\boldsymbol{u}_i^*)_{i\in\mathcal{I}})$ are the corresponding transition frequencies. Recall the single-server $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ problem

$$\begin{aligned}\underset{\overline{N},\ \overline{\sigma},\ \pi}{\text{minimize}} \quad & \overline{N}\\ \text{subject to} \quad & \mathbb{E}\left[h\big(\overline{\boldsymbol{K}}(\infty)\big)\middle|\overline{\boldsymbol{K}}(\infty) \neq \boldsymbol{0}\right] \leq \epsilon,\\ & \overline{N} \cdot \overline{\lambda}_i = \lambda_i r, \quad \forall i \in \mathcal{I}.\end{aligned} \tag{3 revisited}$$

Observe that under the policy $\overline{\sigma}(\boldsymbol{\pi}^*, (\boldsymbol{u}_i^*)_{i\in\mathcal{I}})$, the cost rate of resource contention is $\boldsymbol{h}^T\boldsymbol{\pi} \leq \epsilon(1 - \pi(\boldsymbol{0}))$, the request rate of type i jobs is $\overline{\lambda}_i = \boldsymbol{1}_o^T\boldsymbol{u}_i^* = \Phi^* \cdot \lambda_i$, so

$$\begin{aligned}&\mathbb{E}\left[h\big(\overline{\boldsymbol{K}}(\infty)\big)\middle|\overline{\boldsymbol{K}}(\infty) \neq \boldsymbol{0}\right] = \frac{\boldsymbol{h}^T\boldsymbol{\pi}}{1 - \pi(\boldsymbol{0})} \leq \epsilon,\\ &\overline{\lambda}_i = \Phi^* \cdot \lambda_i, \quad \forall i \in \mathcal{I}.\end{aligned}$$

where we have used the fact that $h(\boldsymbol{0}) = 0$ in the first equality. Therefore, $(\Phi^*/r, \overline{\sigma}(\boldsymbol{\pi}^*, (\boldsymbol{u}_i^*)_{i\in\mathcal{I}}), \boldsymbol{\pi})$ is a feasible solution to the single-server problem $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$, achieving the objective value of r/Φ^*, which is the optimal value because $r/\Phi^* \leq \overline{N}^*$. □

D PERFORMANCE GUARANTEE OF JOIN-REQUESTING-SERVER WITH AN ESTIMATED MODEL

D.1 Assumptions and result

In this section, we consider the performance of JRS when it is based on an estimated model. We will state the performance guarantee in terms of the estimation error, and give a proof sketch by pointing out which part of Theorem 3's proof needs to be changed accordingly.

Consider the setting where the maximal jobs on a server $K_{\max}$, the set of job phases $\mathcal{I}$, the cost rate function $h(\cdot)$, and the budget ϵ are all known. However, we only have estimations of the jobs' arrival rates, internal transition rates, and departure rates.

Specifically, for any $i, i' \in \mathcal{I}$, let $\widetilde{\mu}_{ii'}$ be the *true* rate of internal transition from phase i to phase i'; let $\widetilde{\mu}_{i\perp}$ be the *true* departure rate of phase i; let $\widetilde{\lambda}_i r$ be the *true* arrival rate of type i jobs. We let $\mu_{ii'}$'s, $\mu_{i\perp}$'s, and $\lambda_i r$'s be the *estimated* internal transition rates, departure rates, and arrival rates, respectively. We assume that there exists a small positive constant δ that is independent of the scaling factor r such that the following assumptions hold.

Assumption 1 (δ-accurate estimation)**.** For any $i, i' \in \mathcal{I}$,

$$|\mu_{ii'} - \widetilde{\mu}_{ii'}| \le \delta, \tag{73}$$

$$|\mu_{i\perp} - \widetilde{\mu}_{i\perp}| \le \delta, \tag{74}$$

$$\left|\lambda_i - \widetilde{\lambda}_i\right| \le \delta. \tag{75}$$

Assumption 2 (Scaling of the single-server objective value)**.** Consider the single-server problem in (3). Let $(\overline{N}, \overline{\sigma}, \pi)$ be a solution feasible to the single-server problem with the estimated parameters that are δ-accurate, where $\delta \in [0, \delta_{\max})$, for some constant $\delta_{\max} > 0$. We assume that there exist constants $0 < m_1 < m_2$ independent of δ and r such that

$$m_1 r \le \overline{N} \le m_2 r.$$

Assumption 3 (δ-insensitivity of the optimal value)**.** Consider the single-server problem in (3). Let the optimal value of the single-server problem with the estimated parameters be $\overline{N}^*$, where the estimated parameters are δ-accurate; let the optimal value of the single-server problem with true parameters be $\overline{N}^*_{\text{true}}$. We assume that

$$\overline{N}^* \le \overline{N}^*_{\text{true}} + \delta r.$$

We also assume that JRS can accurately simulate the virtual jobs.

Assumption 4 (Accurate simulation)**.** The virtual jobs simulated in JRS follow the true transition dynamics.

Given the above assumptions, we have the following proposition that states the optimality gap of JRS policy with estimated model parameters, which has a similar form as Theorem 3 for JRS under true model parameters.

Proposition 1 (Optimality gap with model estimation)**.** *Consider a stochastic bin-packing problem in service systems with time-varying job resource requirements. Let the infinite-server policy σ be JRS with subroutine $\overline{\sigma}$. Suppose σ is specified based on an estimated model satisfying Assumption 1, 2, 3, and 4, for δ s.t. $\delta \in [0, \delta_{\max})$, where $\delta_{\max}$ is some positive constant independent of r. Let $\overline{N}$ be the objective value achieved by $\overline{\sigma}$ in the single-server problem $\overline{\mathcal{P}}((\lambda_i r)_{i\in\mathcal{I}}, \epsilon)$ with estimated parameters. Under σ, for any initial state, we have*

$$\left|\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathbb{E}\left[X_{\boldsymbol{k}}\right] - \left\lceil \overline{N} \right\rceil \cdot \mathbb{P}\left(\overline{\boldsymbol{K}} \neq \boldsymbol{0}\right)\right| = O\left(\sqrt{r}\right) + \delta \cdot O\left(r\right), \tag{76}$$

$$\left|\sum_{\boldsymbol{k}\neq\boldsymbol{0}} h(\boldsymbol{k})\mathbb{E}\left[X_{\boldsymbol{k}}\right] - \left\lceil \overline{N} \right\rceil \cdot \mathbb{E}\left[h(\overline{\boldsymbol{K}})\right]\right| = O\left(\sqrt{r}\right) + \delta \cdot O\left(r\right), \tag{77}$$

where $\overline{\boldsymbol{K}}$ denotes the steady-state configurations of the single-server system under $\overline{\sigma}$ with the estimated parameters. If we let $\overline{\sigma}$ be the optimal policy of the single-server problem with estimated parameters, for any initial state, we have

$$N(\sigma) \le \left(1 + B\delta + O\left(r^{-0.5}\right)\right) \cdot \overline{N}^*_{true}, \tag{78}$$

$$C(\sigma) \le \left(1 + B\delta + O\left(r^{-0.5}\right)\right) \cdot \epsilon, \tag{79}$$

where B is a positive constant independent of r. In other words, σ is $\left(1 + B\delta + O\left(r^{-0.5}\right), 1 + B\delta + O\left(r^{-0.5}\right)\right)$-optimal.

Remark 3. Note even if the single-server system with estimated parameters $\boldsymbol{k}^0$-irreducible under $\overline{\sigma}$, it is hard to guarantee that the original system is $\boldsymbol{k}^0$-irreducible due to the estimation errors. Consequently, the steady-state performance metrics $N(\sigma)$ and $C(\sigma)$ could depend on the system's initial state. Fortunately, our proof for Theorem 3 does not rely on the uniqueness of the stationary distribution, so we can adapt the proof to show the inequalities in Proposition 1 for any initial state.

Remark 4. Note that Assumption 4 ensures that the real jobs and virtual jobs on a server are indistinguishable, so that $(\widehat{\boldsymbol{K}}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$ is still a Markov chain. Recall that $\widehat{\boldsymbol{K}}^{\ell}$ and and $\boldsymbol{\eta}^{\ell}$ denote the observed configuration and tokens for each normal server ℓ, respectively (see Section 5.1). We suspect that Assumption 4 can be removed since our proof does not rely too much on $(\widehat{\boldsymbol{K}}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$ being a Markov chain. However, removing the assumption requires a more careful and notationally heavy analysis. We argue that in practice, this assumption is not restrictive, as one can record the traces of the jobs that arrived in the past and resample virtual jobs from those traces.

D.2 Lemmas

In this section, we give a proof sketch for Proposition 1 when the single-server system under $\overline{\sigma}$ *with estimated parameters* is $\boldsymbol{k}^0$-irreducible. The argument of extending to the general case is essentially the same as Appendix B.3, so we omit it here.

On a high level,the proof of Proposition 1 is similar to that of Theorem 1. In particular, the key steps of the proof are to show that $d(\boldsymbol{K}^{1:L}, \overline{\boldsymbol{K}}^{1:L}) = O(\sqrt{r})$ and $\sum_{\ell=L+1}^{\infty} \sum_{i \in \mathcal{I}} K_i^{\ell} = O(\sqrt{r})$ as $r \to \infty$, where $L = \lceil \overline{N} \rceil$, and $\overline{\boldsymbol{K}}^{1:L}$ are i.i.d. copies of steady-state configurations of the single-server system under $\overline{\sigma}$ with the *estimated parameters*, and $\boldsymbol{K}$ is the steady-state configurations of the infinite server system under σ.

Proposition 1 is based on the three lemmas stated below. The proof of Proposition 1 using the three lemmas is essentially the same as the argument in Section 5.3.

Lemma 8. *Under the conditions of Proposition 1 and the single-server system with estimated parameters under $\overline{\sigma}$ being $\boldsymbol{k}^0$-irreducible, for any initial state, we have*

$$d\left(\widehat{\boldsymbol{K}}^{1:L}, \overline{\boldsymbol{K}}^{1:L}\right) = O(\sqrt{r}) + \delta \cdot O(r)$$

Lemma 9. *Under the conditions of Proposition 1 and the single-server system with estimated parameters under $\overline{\sigma}$ being $\boldsymbol{k}^0$-irreducible, for any initial state and $i \in \mathcal{I}$, the steady-state expected number of virtual jobs of type i s.t.*

$$\mathbb{E}\left[\sum_{\ell=1}^{L} \zeta_i^{\ell}\right] = O(\sqrt{r}) + \delta \cdot O(r).$$

Lemma 10. *Under the conditions of Proposition 1 and the single-server system with estimated parameters under $\overline{\sigma}$ being $\boldsymbol{k}^0$-irreducible, for any initial state and $i \in \mathcal{I}$, the steady-state expected number of type i jobs on backup servers s.t.*

$$\mathbb{E}\left[\sum_{\ell=L+1}^{\infty} K_i^{\ell}\right] = O(\sqrt{r}) + \delta \cdot O(r).$$

These three lemmas are analogous to Lemma 2, Lemma 3, and Lemma 4, respectively. In the rest of the section, we sketch the proofs for the three lemmas above, highlighting the difference from the proofs of their analogues.

D.3 Proof sketch for Lemma 8

Recall from Section 5.4 that the proof of Lemma 2 is based on Stein's method, which compares the generator of the i.i.d. copies of the single-server system with the generator of the infinite-server system. To write down the generators with the estimated model, recall that in the single-server system, each transition can be represented by the diagram

$$\boldsymbol{k} \to \boldsymbol{k}' \to \boldsymbol{k}' + \boldsymbol{a},$$

where the arrow $\boldsymbol{k} \to \boldsymbol{k}'$ denotes an internal transition or a departure if $\boldsymbol{k} \neq \boldsymbol{k}'$; the arrow $\boldsymbol{k}' \to \boldsymbol{k}' + \boldsymbol{a}$ denotes a job request that is made right after reaching $\boldsymbol{k}'$. For any $\boldsymbol{k}$, let $E(\boldsymbol{k})$ be the set of $(\boldsymbol{k}', \boldsymbol{a}) \in \mathcal{K}^2$ such that $\boldsymbol{k}' + \boldsymbol{a} \in \mathcal{K}$. We define two sets of transition rates as below: for any $\boldsymbol{k} \in \mathcal{K}$ and $(\boldsymbol{k}', \boldsymbol{a}) \in E(\boldsymbol{k})$,

- Under the estimated parameters and the policy $\overline{\sigma}$, we let $\gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ be the rate of the transition $\boldsymbol{k} \to \boldsymbol{k}' \to \boldsymbol{k}' + \boldsymbol{a}$, and let $\gamma_{\boldsymbol{k}} \triangleq \sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ be the total transition rate at configuration $\boldsymbol{k}$.
- Under the *true parameters and the policy* $\overline{\sigma}$, we let $\widetilde{\gamma}_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ be the rate of the transition $\boldsymbol{k} \to \boldsymbol{k}' \to \boldsymbol{k}' + \boldsymbol{a}$, and let $\widetilde{\gamma}_{\boldsymbol{k}} \triangleq \sum_{\boldsymbol{k}',\boldsymbol{a}} \widetilde{\gamma}_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ be the total transition rate at configuration $\boldsymbol{k}$.

Let $\overline{G}$ be the generator of $L = \lceil \overline{N} \rceil$ i.i.d. copies of single-server systems under the estimated parameters. For any $g\colon \mathcal{K}^L \to \mathbb{R}$, we have

$$\overline{G}g(\boldsymbol{k}^{1:L}) = \sum_{\ell=1}^{L} \sum_{\boldsymbol{k}',\boldsymbol{a}} \gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(g(\cdot, \boldsymbol{k}' + \boldsymbol{a}, \cdot) - g(\cdot, \boldsymbol{k}^\ell, \cdot) \right), \tag{80}$$

where $\sum_{\boldsymbol{k}',\boldsymbol{a}}$ is a shorthand for $\sum_{(\boldsymbol{k}',\boldsymbol{a}) \in E(\boldsymbol{k})}$. Let $\widehat{G}$ be the generator of $(\widehat{\boldsymbol{K}}^{1:L}(t), \boldsymbol{\eta}^{1:L}(t))$ for the infinite-server system. For any function $g\colon \mathcal{K}^L \to \mathbb{R}$ and $\psi(\boldsymbol{k}^{1:L}, \boldsymbol{\eta}^{1:L}) = g(\boldsymbol{k}^{1:L} + \boldsymbol{\eta}^{1:L})$, we have

$$\begin{aligned} \widehat{G}\psi(\boldsymbol{k}^{1:L}, \boldsymbol{\eta}^{1:L}) = &\sum_{\ell=1}^{L} \sum_{\boldsymbol{k}',\boldsymbol{a}} \widetilde{\gamma}_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(g(\cdot, \boldsymbol{k}' + \boldsymbol{a}, \cdot) - g(\cdot, \boldsymbol{k}^\ell, \cdot) \right) \mathbb{1}_{\{\boldsymbol{\eta}^\ell = \boldsymbol{0}\}} \\ &+ \sum_{\ell=1}^{L} \sum_{\boldsymbol{k}',\boldsymbol{a}} \widetilde{\gamma}_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})} \left(g(\cdot, \boldsymbol{k}' + \boldsymbol{\eta}^\ell, \cdot) - g(\cdot, \boldsymbol{k}^\ell + \boldsymbol{\eta}^\ell, \cdot) \right) \mathbb{1}_{\{\boldsymbol{\eta}^\ell \neq \boldsymbol{0}\}}. \end{aligned} \tag{81}$$

To prove Lemma 8, we need to show that for any $f \in \mathrm{Lip}(1)$, $\boldsymbol{k}^{1:L} \in \mathcal{K}^L$, and $\boldsymbol{\eta}^{1:L} \in \mathcal{K}^L$,

$$\overline{G}g_f(\boldsymbol{k}^{1:L} + \boldsymbol{\eta}^{1:L}) - \widehat{G}\psi_f(\boldsymbol{k}^{1:L}, \boldsymbol{\eta}^{1:L}) = O\big(\sqrt{r}\big) + \delta \cdot O\big(r\big), \tag{82}$$

where g_f is the solution to

$$\mathbb{E}\left[f\big(\overline{\boldsymbol{K}}^{1:L}\big) \right] - f\big(\boldsymbol{k}^{1:L}\big) = \overline{G}g_f\big(\boldsymbol{k}^{1:L}\big), \tag{83}$$

and $\psi_f(\boldsymbol{k}^{1:L}, \boldsymbol{\eta}^{1:L}) = g_f(\boldsymbol{k}^{1:L} + \boldsymbol{\eta}^{1:L})$. Same as the proof of Lemma 2, we prove (82) in two steps: the generator comparison step, and the Stein factor bound step.

In the generator comparison step, we observe that the formula of $\overline{G}g(\boldsymbol{k}^{1:L})$ and $\widehat{G}\psi(\boldsymbol{k}^{1:L}, \boldsymbol{\eta}^{1:L})$ in (80) and (81) look almost the same as (15) and (17), except that the rates in (81) are $\widetilde{\gamma}_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})}$ instead of $\gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})}$. As a result, after carrying out similar calculations as in the poof of Lemma 2, we get an extra error term involving $\gamma_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})} - \widetilde{\gamma}_{\boldsymbol{k}^\ell,(\boldsymbol{k}',\boldsymbol{a})}$, which can be bounded using the lemma below. This error term results in the $\delta \cdot O(r)$ in (82).

Lemma 11. *Under Assumption 1, for any $\boldsymbol{k} \in \mathcal{K}$ and $(\boldsymbol{k}', \boldsymbol{a}) \in E(\boldsymbol{k})$, we have*

$$\left|\widetilde{\gamma}_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})} - \gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}\right| \le K_{\max}\delta. \tag{84}$$

Proof. When $\boldsymbol{k} = \boldsymbol{k}'$, $\gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ and $\widetilde{\gamma}_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ are both the rate of adding $\boldsymbol{a}$ jobs via a proactive request under the policy $\overline{\sigma}$, so they are identical.

When $\boldsymbol{k} \neq \boldsymbol{k}'$ and $\boldsymbol{a} = \boldsymbol{0}$, $\gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ is equal to the rate of going from $\boldsymbol{k}$ to $\boldsymbol{k}'$ via an internal transition or departure under the estimated job model, while $\widetilde{\gamma}_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ is under the true job model. Because there are at most $K_{\max}$ jobs, and by our assumption the estimation error of each job's transition rates are bounded by δ, thus $\gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ and $\widetilde{\gamma}_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ differ by at most $K_{\max}\delta$.

When $\boldsymbol{k} \neq \boldsymbol{k}'$ and $\boldsymbol{a} \neq \boldsymbol{0}$, $\gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ is equal to the rate of going from $\boldsymbol{k}$ to $\boldsymbol{k}'$ via an internal transition or departure, multiplied by the probability of adding $\boldsymbol{a}$ jobs via a reactive request, under the estimated job model and the policy $\overline{\sigma}$; $\widetilde{\gamma}_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ is under the true job model instead of the estimated job model, but uses the same policy. Because the rate of going from $\boldsymbol{k}$ to $\boldsymbol{k}'$ differs by at most $K_{\max}\delta$ under two different job models, and the probability of adding $\boldsymbol{a}$ jobs after going from $\boldsymbol{k}$ to $\boldsymbol{k}'$ is the same under the same policy, thus $\gamma_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ and $\widetilde{\gamma}_{\boldsymbol{k},(\boldsymbol{k}',\boldsymbol{a})}$ differ by at most $K_{\max}\delta$. □

In the Stein factor bound step, we need to show that

$$\sup_{\boldsymbol{k},\boldsymbol{k}'\in\mathcal{K}} \left|g_f(\cdot,\boldsymbol{k}',\cdot) - g_f(\cdot,\boldsymbol{k},\cdot)\right| = O(1). \tag{85}$$

This involves analyzing the i.i.d. copies of the single-server system, and the fact that the single-server system with estimated parameters is $\boldsymbol{k}^0$-irreducible under $\overline{\sigma}$. This part of the proof is identical to the corresponding part of the proof of Lemma 2.

D.4 Proof sketch for Lemma 9 and Lemma 10

The proof of Lemma 9 and Lemma 10 has a similar structure as the proof of Lemma 3 and Lemma 4. In the first step, we use Little's law to bound expectations of the number of type i virtual jobs, V_i, and the number of type i jobs on backup servers, Y_i. We have two equations almost the same as (36) and (37) except that the rates $\gamma_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}$ and $\lambda_i r$ are replaced by $\widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}$ and $\widetilde{\lambda}_i r$:

$$\mathbb{E}[V_i] \le t_{\max}\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} dv_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right], \tag{86}$$

$$\mathbb{E}[Y_i] \le t_{\max}\mathbb{E}\left[\widetilde{\lambda}_i r \cdot dy_i\right], \tag{87}$$

where $t_{\max}$ is the maximal expected service time of any type of virtual job or real job. Because $t_{\max} = O(1)$, it suffices to bound $\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} dv_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right]$ and $\mathbb{E}\left[\widetilde{\lambda}_i r \cdot dy_i\right]$.

In the second step, we utilize the fact that the two Lyapunov functions $g(z_i) = z_i$ and $g(z_i) = z_i^2$ have zero drift in steady-state, where z_i is the total number of type i tokens. We get two equalities similar to (38) and (41):

$$\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}(a_i - dv_i)\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} + \widetilde{\lambda}_i r(-1 + dy_i)\right] = 0. \tag{88}$$

$$\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} dv_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right] \tag{89}$$

$$= \frac{1}{\eta_{\max}} \cdot \mathbb{E}\left[\left(\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}} \widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} a_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}} - \widetilde{\lambda}_i r\right) \cdot Z_i\right] \tag{90}$$

$$+\frac{1}{2\eta_{\max}}\cdot\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}\left(a_i^2-(dv_i)^2\right)\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}+\widetilde{\lambda}_i r\cdot(1-(dy_i)^2)\right]. \tag{91}$$

In the third step, we use the above two equalities to bound $\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}dv_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right]$ and $\mathbb{E}\left[\widetilde{\lambda}_i r\cdot dy_i\right]$. We first use the equality in (89) to (91) to bound $\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}dv_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right]$. Following the same argument as the proof of Lemma 3 and Lemma 4 until (47), we can show that

$$(91)\le O(\sqrt{r}),$$

$$\begin{aligned}(90)&\le\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}-\widetilde{\lambda}_i r\right|\right]\\&\le\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i-\widetilde{\lambda}_i r\right|\right]+O(\sqrt{r})\\&\le\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\widetilde{\gamma}_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i-\widetilde{\lambda}_i r\right|\right]+O(\sqrt{r})+\delta\cdot O(r),\end{aligned} \tag{92}$$

where to get (92), we apply Lemma 8 to replace $\widehat{\boldsymbol{K}}^\ell$ with $\overline{\boldsymbol{K}}^\ell$, which causes an $O(\sqrt{r})+\delta\cdot O(r)$ error. Next, we show that

$$\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\widetilde{\gamma}_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i-\widetilde{\lambda}_i r\right|\right]=O(\sqrt{r})+\delta\cdot O(r). \tag{93}$$

By Assumption 1 and Lemma 11, we have

$$\left|\widetilde{\lambda}_i r-\lambda_i r\right|\le\delta r, \tag{94}$$

$$\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\widetilde{\gamma}_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i-\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i\right|\le\delta\cdot O(r). \tag{95}$$

These two bounds allow us to replace $\widetilde{\gamma}_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}$ and $\widetilde{\lambda}_i r$ on the LHS of (93) with $\gamma_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}$ and $\lambda_i r$ at the cost of introducing $\delta\cdot O(r)$ error. Moreover, because $\{\overline{\boldsymbol{K}}^\ell\}_{\ell=1,\dots,L}$ are i.i.d., $\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i$ concentrates around its mean with $O(\sqrt{r})$ error, where the mean can be shown to be

$$\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i\right]=\overline{\lambda}_i L=\lambda_i r+O(1).$$

Note that the first equality in (96) holds because for each ℓ, $\mathbb{E}[\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i]$ is equal to $\overline{\lambda}_i$, i.e., the long-run average rate of requesting type i jobs on a single-server system with estimated parameters under $\overline{\sigma}$; the second equality in (96) is because $L=\lceil\overline{N}\rceil$, and $\overline{N}\overline{\lambda}_i=\lambda_i r$. As a result,

$$\mathbb{E}\left[\left|\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\gamma_{\overline{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}a_i-\lambda_i r\right|\right]\le O(\sqrt{r}). \tag{96}$$

Combining (94) to (96), we get (93). Therefore,

$$\mathbb{E}[V_i]\le t_{\max}\mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{\boldsymbol{k}',\boldsymbol{a}}\widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})}dv_i\mathbb{1}_{\{\boldsymbol{\eta}^\ell=\boldsymbol{0}\}}\right]$$

$$\leq O(1) \cdot \big((90) + (91)\big)$$
$$\leq O\big(\sqrt{r}\big) + \delta \cdot O\big(r\big),$$

which completes the proof of Lemma 9.

Finally, it is straightforward to show that $\mathbb{E}\left[\widetilde{\lambda}_i r \cdot dy_i\right] = O(\sqrt{r}) + \delta \cdot O(r)$ using (88) and the bound on $\mathbb{E}\left[\sum_{\ell=1}^{L} \sum_{\boldsymbol{k}',\boldsymbol{a}} \widetilde{\gamma}_{\widehat{\boldsymbol{K}}^\ell,(\boldsymbol{k}',\boldsymbol{a})} dv_i \mathbb{1}_{\{\boldsymbol{\eta}^\ell = \boldsymbol{0}\}}\right]$. Therefore, $\mathbb{E}[Y_i] \leq t_{\max} \mathbb{E}\left[\widetilde{\lambda}_i r \cdot dy_i\right] = O(\sqrt{r}) + \delta \cdot O(r)$. This proves Lemma 10.

Received August 2023; revised October 2023; accepted October 2023