# Sharp Waiting-Time Bounds for Multiserver Jobs

Yige Hong
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Weina Wang
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

## ABSTRACT

Multiserver jobs, which are jobs that occupy multiple servers simultaneously during service, are prevalent in today's computing clusters. But little is known about the delay performance of systems with multiserver jobs. We consider queueing models for multiserver jobs in a *scaling regime* where the system load becomes heavy and meanwhile the total number of servers in the system and the number of servers that a job needs become large. Prior work has derived upper bounds on the queueing probability in this scaling regime. However, without proper lower bounds, the existing results cannot be used to differentiate between policies. In this paper, we study the delay performance by establishing *sharp bounds* on the *mean waiting time* of multiserver jobs, where the waiting time of a job is the time spent in queueing rather than in service. We first consider the commonly used First-Come-First-Serve (FCFS) policy and characterize the *exact order* of its mean waiting time. We then prove a lower bound on the mean waiting time of all policies, and demonstrate that there is an *order gap* between this lower bound and the mean waiting time under FCFS. We finally complement the lower bound with an *achievability* result: we show that under a priority policy that we call P-Priority, the mean waiting time achieves the order of the lower bound. This achievability result implies the tightness of the lower bound, the asymptotic optimality of P-Priority, and the strict suboptimality of FCFS.

## CCS CONCEPTS

• **Mathematics of computing → Queueing theory**; **Markov processes**; • **Networks → Network performance analysis**.

**ACM Reference Format:**
Yige Hong and Weina Wang. 2022. Sharp Waiting-Time Bounds for Multiserver Jobs. In *The Twenty-third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '22), October 17–20, 2022, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3492866.3549717

## 1 INTRODUCTION

In today's large-scale computing clusters behind cloud platforms, *multiserver jobs* have become increasingly prevalent, where a multiserver job is a job that demands to occupy multiple "servers" (which can be multiple physical servers, multiple CPU cores, etc.) simultaneously during its runtime [2, 20, 31, 34]. For example, cloud platforms allow users to specify the number of CPU cores in their virtual machines or containers, and this information can be utilized by centralized schedulers to make scheduling decisions (see, for example, [1, 34]). Moreover, the number of "servers" that a multiserver job requests, which we refer to as the *server need*, is becoming increasingly large. This trend is driven by machine learning jobs from applications like TensorFlow [2], where the jobs are highly parallel and require synchronization. According to the statistics from Google's Borg Scheduler [34], the server needs in Borg can vary across six orders of magnitudes.

In this paper, we study the impact of multiserver jobs on the delay performance of large-scale computing systems using queueing models. Queueing models with multiserver jobs have been studied in the literature, but quantifying the delay performance is notoriously hard. Exact steady-state distributions can only be derived in highly simplified settings with two servers [10, 13], while the majority of prior work has focused on characterizing stability conditions [3, 14, 27, 29]. However, even for stability, exact conditions are known only for the special cases where all jobs have the same service rates or where there are two job classes.

A recent advance in understanding the delay of multiserver jobs is a characterization of the *queueing probability* in a large system by Wang et al. [36], where the queueing probability is the probability that an arriving job has to queue rather than entering service immediately. Specifically, Wang et al. [36] consider a multiserver job system with $n$ servers, and study the asymptotic scaling regimes where $n$ becomes large. The scaling regimes allow different job types to have different arrival rates, server needs and service rates. Among those parameters, server needs and arrival rates can scale up with $n$. Such scaling regimes capture the trend that different multiserver jobs can be highly heterogeneous, especially in terms of server needs. Wang et al. [36] establish an upper bound on the queueing probability, based on which they give a sufficient condition for the queueing probability to diminish as $n$ goes to infinity.

Although the work [36] identifies when the queueing probability diminishes in large systems, which is a much desirable operating scenario, it does not provide much insight for differentiating between scheduling policies. In particular, the queueing probability upper bound in [36] holds for any scheduling policy that is reasonably work-conserving (although the bound is presented only for the First-Come-First-Serve policy in [36]). Moreover, queueing probability does not directly translate to delay of jobs.

In this paper, we focus on the *waiting time* of jobs, which is the time a job spends waiting in the queue (not receiving any service), under various scheduling policies. The waiting time is a performance metric that is directly related to job delay. Our goal is to establish bounds on the mean waiting time that are *order-wise tight* as the number of servers, $n$, scales. Such tight bounds will enable us to differentiate between policies based on their delay performance. We comment that there has been a line of work in the literature [21–24, 32, 37, 38] that focuses on quantifying when the

---

*MobiHoc '22, October 17–20, 2022, Seoul, Republic of Korea*

ACM ISBN 978-1-4503-9165-8/22/10.
https://doi.org/10.1145/3492866.3549717

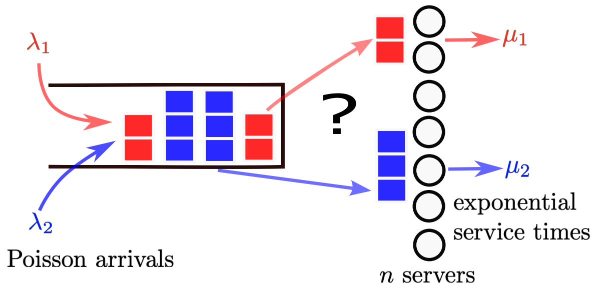


**Figure 1: A multiserver-job system with two types of jobs. Type 1 jobs have arrival rate $\lambda_1$, service rate $\mu_1$, and server need $\ell_1 = 2$. Type 2 jobs have arrival rate $\lambda_2$, service rate $\mu_2$, and server need $\ell_2 = 3$.**

mean waiting time diminishes in large systems for various queueing models. However, little is known on *how fast* the mean waiting time diminishes due to the lack of lower bounds. Our results provide the rate of diminishing when the mean waiting time does diminish, but our tight bounds on the mean waiting time are not limited to the "diminishing" scenario.

Since the First-Come-First-Serve (FCFS) policy is widely used as a default policy in practice and also it receives the most attention from theoretical studies of multiserver jobs [3, 10, 13, 14, 27, 29], we will first examine FCFS and understand the exact order of the mean waiting time under it. Then a natural question that arises is: *can any policy outperform FCFS in terms of the mean waiting time?* More generally, we aim to answer the following fundamental questions:

- *What is the optimal order of the mean waiting time as the system scales?*
- *Which policy achieves the optimal order?*

## 1.1 Model and performance metric

We consider a system that consists of $n$ servers and $I$ types of jobs. An example is illustrated in Figure 1. Suppose type $i$ jobs needs the simultaneous service of $\ell_i$ servers. We sort the job types so that their *server needs* $\ell_i$'s satisfy $\ell_1 \le \ell_2 \le \dots \ell_I$. Let the *maximal server need* $\ell_{\max}$ to be $\ell_{\max} = \max_{i\in\{1,2,\dots,I\}} \ell_i = \ell_I$, and we call type $I$ jobs the *maximal-need jobs.*

The dynamics of the system are as follows. For each $i = 1, 2, \dots, I$, type $i$ jobs arrive to the system following a Poisson process with *arrival rate* $\lambda_i$. Upon arrival, a job either starts service immediately or waits in a centralized queue. When the type $i$ job starts service, it leaves the queue and makes exclusive use of $\ell_i$ servers. The job leaves the system after receiving enough service. The service time of a type $i$ job follows an exponential distribution with *service rate* $\mu_i$. The service times and arrival events are independent.

During the operation of the system, a scheduling policy is used to determine which set of jobs to serve at each time. The scheduling policy is allowed to be preemptive, i.e., we can put a job in service back to the queue and resume its service later.

We measure the performance of our scheduling policy based on *mean waiting time* as defined below: let $T_i^w(\infty)$ denote the waiting time of type $i$ jobs in steady-state, then the mean waiting time is defined as the steady-state expected waiting time averaged over all job types, i.e.,

$$\mathbb{E}\left[T^w(\infty)\right] = \frac{1}{\lambda}\sum_{i=1}^{I} \lambda_i \mathbb{E}\left[T_i^w(\infty)\right],$$

where $\lambda \triangleq \sum_{i=1}^{I} \lambda_i$ is the total arrival rate.

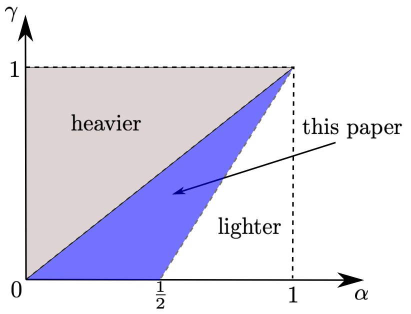


**Figure 2: Scaling regimes under the special parameterization with slack capacity $\delta = n^\alpha$ and maximum server need $\ell_{\max} = n^\gamma$. The traffic is heavier as we move to the upper left. The three triangles are partitioned by lines $\gamma = \alpha$ and $\alpha = \frac{1+\gamma}{2}$.**

## 1.2 Scaling regimes

We study job delay in scaling regimes where the number of servers, $n$, goes to infinity. Specifically, we consider a sequence of systems with parameters scaling up jointly with $n$, and analyze the growth/decrease rate of mean waiting times. In the considered scaling regimes, the arrival rates $\lambda_i$ and server needs $\ell_i$ are allowed to scale with $n$, while the service rate $\mu_i$ and the number of job types $I$ stay constant. One key parameter for specifying a scaling regime is the *slack capacity* $\delta$, defined as $\delta = n - \sum_{i=1}^{I} \frac{\lambda_i \ell_i}{\mu_i}$, which is the expected number of idle servers in steady state. Slack capacity is used to specify the heaviness of traffic, which is alternatively specified by *load* $\rho$ given by $\rho = \sum_{i=1}^{I} \frac{\lambda_i \ell_i}{n\mu_i}$ in literature.

For expositional purposes, here we parameterize the scaling regimes in the following way: $\ell_{\max} = n^\gamma, \delta = n^\alpha$ for some exponents $0 \le \alpha, \gamma < 1$. We assume that the total arrival rate $\lambda = \Theta(n)$. Note that the scaling regimes we study in this paper are more general, and we refer the readers to Section 2 for full details. We aggregate all scaling regimes with the same $(\alpha, \gamma)$ pair into one point and plot out all such points, as shown in Figure 2. We partition the set of exponent pairs $(\alpha, \gamma) \in [0, 1)^2$ into three triangles, using the lines $\alpha = \gamma$ and $\alpha = \frac{1+\gamma}{2}$. The corresponding scaling regimes to the upper left are in general "heavier" than the scaling regimes to the lower right, since the former regimes have larger work variability and smaller slack capacity. We comment that the point $(\alpha, \gamma) = (\frac{1}{2}, 0)$ is analogous to the celebrated Halfin-Whitt regime [15] and $(\alpha, \gamma) = (0, 0)$ is analogous to the Non-Degenerate Slowdown (NDS) regime [7] in traditional multiclass M/M/$n$ models.

We focus on the scaling regimes where $\gamma < \alpha < \frac{1+\gamma}{2}$, marked in blue in Figure 2. The scaling regimes satisfying the condition are not too light; the lighter regimes marked in white, studied in [36], can be shown to have both queueing probability and mean waiting time diminish at a rate faster than any polynomial in $n$ under any reasonably work-conserving policy. Meanwhile, the regimes under study are not too heavy either; the system still enjoys diminishing mean waiting time together with high system utilization.

### 1.3 Results

We present our main results here in the specialized scaling regimes for exposition purposes. General forms with fully specified assumptions are presented in Sections 2 and 3.

Our results and analysis heavily use the asymptotic notation. [1]

- **Mean waiting time under FCFS.** The exact order of the mean waiting time under FCFS is given by
$$\mathbb{E}\left[T^w(\infty)\right]^{\text{FCFS}} = \Theta\left(n^{\gamma-\alpha}\right). \tag{1}$$
- **Mean waiting time lower bound.** Under any policy, the mean waiting time is lower bounded as
$$\mathbb{E}\left[T^w(\infty)\right] = \Omega\left(n^{-\alpha}\right), \tag{2}$$
- **Order-wise optimal policy.** Consider a static priority policy that we call the *P-Priority policy*, which preemptively prioritizes the jobs with smaller server needs. Then the mean waiting time under P-Priority achieves the lower bound in (2), i.e.,
$$\mathbb{E}\left[T^w(\infty)\right]^{\text{P-Priority}} = \Theta\left(n^{-\alpha}\right). \tag{3}$$
Therefore, the P-Priority policy is order-wise optimal in the mean waiting time.

Comparing the mean waiting time under FCFS and under P-Priority, we can see that FCFS is strictly suboptimal, and P-Priority improves upon FCFS by a factor of $\Theta(n^{\gamma})$.

A key to proving the mean waiting time results above is the *order-wise tight* bounds on the expected workload we establish (Lemma 1 and Lemma 2). In addition, although we consider the system under the traffic regime where $\gamma < \alpha < \frac{1+\gamma}{2}$, we still need to analyze "subsystems" that are in the lighter traffic regime. In this lighter regime, we show that the total server need decays faster than any polynomial (Lemma 6). All these lemmas hold under a very general class of policies, so they could be potentially relevant when we study policies other than FCFS and P-Priority.

***Results on queueing probability.*** As a by-product to our analysis, we further derive an upper bound on the queueing probability under a work-conserving policy, presented in Corollary 2, which significantly improves upon the queueing probability upper bound in [36] in a slightly more constrained traffic regime.

***Simulation experiments.*** The P-Priority policy we consider in our analysis is a *preemptive* policy, but preemption is usually not preferable in practice. Therefore, we use simulation experiments to explore a *non-preemptive* version of the priority policy, which we call the N-Priority policy. N-Priority serves a job with the smallest server need in the queue when enough number of servers free up. Our simulation experiments compare the mean waiting time under FCFS, P-Priority, and N-Priority. The simulation results, presented in Section 9, show that N-Priority has comparable performance with P-Priority, and demonstrate the performance gap between FCFS and P-Priority/N-Priority.

### 1.4 Technical challenges

The main technical challenges in analyzing the considered multiserver-job system are rooted in the *heterogeneity* among job types in both their service rates and their server needs. Such heterogeneity makes the system dynamics multidimensional: neither the total number of jobs in service nor the total number of busy servers determines the current job departure rate. We comment that even for a classical multi-class M/M/$n$ system, where there are multiple job types with different service rates but all job types have a server need of 1, finding an optimal scheduling policy is known to be a hard problem, and solutions are available mostly in the so-called Halfin-Whitt heavy-traffic regime through the diffusion control problem [6, 8, 17]. Compared with the classical multi-class M/M/$n$ system, our multiserver-job system has an additional layer of intricacy due to the heterogeneous server needs, which makes it possible for the system to have servers idling while there are jobs in the queue.

To address the challenges due to heterogeneity, our analysis relies on various state-space concentration results. State-space concentration is a phenomenon where the state concentrates around a *subset* of the state space in steady state, observed in queueing systems in heavy-traffic or large-system regimes [21–24, 35, 38]. In the multiserver-job system we consider, state-space concentration results are crucial for analyzing the system dynamics when the queue is nonempty. The scenario when the queue is nonempty is especially important to our scaling regimes since the queueing probability may not be diminishing even when the mean waiting time is diminishing. This contrasts with the analysis in prior work [36], which focuses on diminishing queueing probability. Furthermore, our performance goal is to achieve the *optimal order* of the mean waiting time in large systems, which deviates from the traditional performance goal of minimizing delay or certain long-run cost.

### 1.5 Additional related work

We provide some additional references to models that are related to the multiserver-job model we consider. One model is the dropping model [4, 9, 19, 30, 33, 39], which is a lossy version of our model. The dropping model considers a system that drops the incoming jobs when there are no available servers, with the goal of minimizing the dropping rate. The virtual machine (VM) scheduling problem studied in [25, 28, 40] considers a system with multiple servers, each with a certain units of resources. A VM job requests multiple units of resources and can only be placed on a single server. Most existing work for VM scheduling focuses on stability and only limited results are available for delay. Another class of models consider jobs consisting of multiple tasks that can run on multiple servers but do not require simultaneous execution. Examples of such models include the parallel task model (also called the fork-join model; see, e.g., [37, 41]) and the batch arrival model (see, e.g., [11]).

## 2 MODEL

A basic description of the system parameters and dynamics has been given in the introduction. In this section, we provide formal

---

[1] We use the standard Bachmann–Landau notation. Consider two sequences $a(n)$ and $b(n)$ (or simply $a$ and $b$), where $b(n)$ is positive for large enough $n$. Then $a = O(b)$ if $\limsup_{n\to\infty} \frac{|a|}{b} < \infty$; $a = o(b)$ if $\lim_{n\to\infty} \frac{a}{b} = 0$; $a = \Omega(b)$ if $\liminf_{n\to\infty} \frac{a}{b} > 0$, which is equivalent to $b = O(a)$; $a = \omega(b)$ if $\lim_{n\to\infty} \frac{|a|}{b} = \infty$, which is equivalent to $b = o(a)$; $a = \Theta(b)$ if $a$ satisfies both $a = O(b)$ and $a = \Omega(b)$.

descriptions of the scheduling policies, the system states, the scaling regimes, and the concept of subsystems used in our analysis.

***Scheduling policies.*** A scheduling policy decides which jobs to put into service at any moment of time. We are interested in the following two policies:

- *First-Come-First-Serve (FCFS)*: Jobs are placed onto servers in a First-Come-First-Serve fashion until either the next job in queue does not fit or all the jobs are in service.
- *Preemptive Priority (P-Priority)*: Recall that the job types are indexed in a way such that $\ell_1 \le \ell_2 \le \cdots \le \ell_I$. We assign priorities to job types such that a smaller index has a higher priority. Whenever there is a job arrival or departure, P-Priority preempts all the jobs in service and determines a new schedule from scratch. P-Priority starts from job type 1 and places as many type 1 jobs as possible onto servers. After this, if there are still servers available, P-Priority goes to the next priority level, type 2, and places as many type 2 jobs as possible onto servers. This procedure continues until no more jobs in the queue can fit into the servers.

***System state.*** Under FCFS or P-Priority, a Markovian representation of the system state can be described as follows. The state $\boldsymbol{u}$ of the Markov chain is an ordered list of the jobs in the system, sorted in their order of arrival, and each entry of $\boldsymbol{u}$ describes the type of the corresponding job and whether the job is in service or not. Let the state space be denoted as $\mathcal{U}$. Although the state space is infinite dimensional, in our analysis, we typically only need to focus on three $I$-dimensional vectors defined below.

For any time $t$ and each job type $i$, let $X_i(t)$ denote the number of type $i$ jobs in the system, $Z_i(t)$ denote the number of type $i$ jobs in service, and $Q_i(t) \triangleq X_i(t) - Z_i(t)$ denote the number of type $i$ jobs waiting in the queue. Note that since the total number of servers in use cannot exceed $n$, and we cannot serve more jobs than there are in the system, we have the following constraints:

$$\begin{aligned} \sum_{i=1}^{I} \ell_i Z_i(t) \le n \quad \text{for all } t \ge 0, \\ Z_i(t) \le X_i(t) \quad \text{for all } t \ge 0, i \in [I], \end{aligned} \tag{4}$$

where $[I]$ denotes the index set $\{1, 2, \ldots, I\}$.

Let $X_i(\infty)$, $Z_i(\infty)$, and $Q_i(\infty)$ be random variables that follow the corresponding steady-state distributions when they exist. We sometimes use vector representations of these quantities for convenience. For example, we write $\boldsymbol{X}(t) = (X_1(t), X_2(t), \ldots, X_I(t))$. We define the vectors $\boldsymbol{Z}(t)$, $\boldsymbol{Q}(t)$, $\boldsymbol{X}(\infty)$, $\boldsymbol{Z}(\infty)$, and $\boldsymbol{Q}(\infty)$ in a similar way. Note that these random elements correspond to the $n$ server system and thus their distributions depend on $n$. Throughout this paper, for conciseness, we often omit the $(\infty)$ in the steady-state random elements except in theorem or lemma statements.

Recall that our performance metric is the mean waiting time $\mathbb{E}\left[T^w(\infty)\right]$, given by

$$\mathbb{E}\left[T^w(\infty)\right] = \frac{1}{\lambda} \sum_{i=1}^{I} \lambda_i \mathbb{E}\left[T_i^w(\infty)\right],$$

where $T_i^w(\infty)$ is the waiting time of type $i$ jobs in steady-state. Note that by Little's law, the mean waiting time can be written as

$$\mathbb{E}\left[T^w(\infty)\right] = \frac{1}{\lambda} \sum_{i=1}^{I} \mathbb{E}[Q_i(\infty)].$$

Therefore, bounding the mean waiting time reduces to bounding the expected total queue length.

***Scaling regimes.*** Recall that we consider scaling regimes where number of servers, $n$, goes to infinity, and the arrival rates $\lambda_i$ and server needs $\ell_i$ are allowed to scale with $n$, *while the service rate $\mu_i$ and the number of job types $I$ stay constant.* The scaling regimes are specified by the slack capacity $\delta \triangleq n - \sum_{i=1}^{I} \frac{\lambda_i \ell_i}{\mu_i}$, the maximal server need $\ell_{\max} \triangleq \max_{i \in [I]} \ell_i = \ell_I$ and another parameter called the *work variability*: $\sigma^2 \triangleq \sum_{i=1}^{I} \frac{\lambda_i \ell_i^2}{\mu_i^2}$. Work variability reflects the variability of the "work" caused by job arrivals in terms of server–time product, which is $\frac{\ell_i}{\mu_i}$ in expectation for each type $i$ job. To help later presentation, we also define the *load brought by type i jobs $\rho_i$* as $\rho_i = \frac{\lambda_i \ell_i}{n \mu_i}$.

We state our assumptions below. Note that throughout this paper, $\log n$ denotes natural logarithm.

Assumption 1 (Heavy traffic assumption). *The slack capacity $\delta$ is small compared with $\sqrt{\sigma^2}$:*

$$\delta = o\left(\frac{\sqrt{\sigma^2}}{\log n}\right). \tag{5}$$

Assumption 2 (Maximal server need assumption). *There exists a constant $\epsilon_0$ with $0 < \epsilon_0 < 1$ such that*

$$\ell_{\max} \le \epsilon_0 \delta. \tag{6}$$

Assumption 3 (Commonness assumption). *The load brought by the maximal-need jobs is not too small:*

$$\rho_I \triangleq \frac{\lambda_I \ell_I}{n \mu_I} = \omega\left(\sqrt{\frac{\delta \log n}{\sqrt{\sigma^2}} \cdot \frac{\ell_{\max}}{n}} \log n\right). \tag{7}$$

Assumption 1 guarantees that the traffic is not too light, while Assumption 2 guarantees that the system is stable under FCFS and P-Priority. In the simplified setting of Section 1 where $\ell_{\max} = n^\gamma$ and $\delta = n^\alpha$, the first two assumptions correspond to $\alpha < \frac{1+\gamma}{2}$ and $\alpha > \gamma$, which exclude the white and grey parts in Figure 2, respectively. Assumption 3 states that the load brought by the maximal-need jobs are not too small. To understand the right hand side expression in Assumption 3, note that it is automatically satisfied when $\rho_I = \omega\left(\sqrt{\ell_{\max}/n} \log n\right)$. For example, when $\ell_{\max} = \Theta(\sqrt{n})$, then it suffices to have $\rho_I = \omega\left(n^{-1/4} \log n\right)$. However, when the traffic becomes heavier, i.e., when $\frac{\delta \log n}{\sqrt{\sigma^2}}$ becomes smaller, Assumption 3 in (7) can be much weaker than $\rho_I = \omega\left(\sqrt{\ell_{\max}/n} \log n\right)$.

To have an intuitive view of the magnitudes of the parameters, we give the following asymptotics: $\sigma^2 = O(n \ell_{\max})$, $\delta = o\left(n/(\log n)^2\right)$, and $\ell_{\max} \le \epsilon_0 \delta = o\left(n/(\log n)^2\right)$. They can be verified using the definitions and assumptions.

***Subsystems.*** In our analysis, we frequently use the concept of the $i$-th *subsystem*, which is the system that has all type $j$ jobs in the original system with $j \leq i$ and removes all type $k$ jobs with $k \geq i$. In the $i$-th subsystem, the slack capacity becomes $\delta_i = n - \sum_{j=1}^{i} \frac{\lambda_j \ell_j}{\mu_j}$, and the work variability becomes $\sigma_i^2 = \sum_{j=1}^{i} \frac{\lambda_j \ell_j^2}{\mu_j^2}$. Note that $\delta = \delta_I$ and $\sigma_I^2 = \sigma^2$. The maximal server need in the $i$-th system is $\ell_i$ since $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_i$.

As $i$ increases, the load of the $i$-th subsystem gets heavier since $\delta_i$ becomes smaller. There is a *critical index* $i^*$ such that

$$i^* = \min\left\{ i \in [I] \;\middle|\; \delta_i = o\left( \frac{\sqrt{\sigma_i^2}}{\log n} \right) \right\}, \tag{8}$$

i.e., the $i^*$th subsystem is the smallest subsystem whose traffic regime is as heavy as that of the original system. Because of the assumption that $\delta = o\left( \frac{\sqrt{\sigma^2}}{\log n} \right)$, the set in (8) contains at least the index $I$ and $i^*$ is well-defined. Note that $\delta_i$ is monotonically decreasing while $\sigma_i^2$ is monotonically increasing. Thus the index $i^*$ serves as a division point: for any $i$ with $i^* \leq i \leq I$, we have $\delta_i = o\left( \frac{\sqrt{\sigma_i^2}}{\log n} \right)$, resulting in a lighter traffic regime; and for any $i$ with $1 \leq i < i^*$, we have $\delta_i = \Omega\left( \frac{\sqrt{\sigma_i^2}}{\log n} \right)$, resulting in a heavier traffic regime.

## 3 MAIN RESULTS

In this section, we first present our main results under the scaling regimes we specify in Section 2 as Theorems 1, 2, and 3. Then, to demonstrate our results in a more intuitive fashion, we consider the parameterized scaling regimes defined in Section 1.2 as a special case, and present the specialized form of our results as Corollary 1.

Theorem 1 (Mean waiting time under FCFS). *Consider the multiserver-job system with n servers satisfying Assumptions 1 and 2. Under the FCFS policy, for each $i \in [I]$, the expected waiting time of type i jobs satisfies*

$$\mathbb{E}\left[T_i^w(\infty)\right]^{\text{FCFS}} \geq \frac{\sigma^2}{n(\delta + \ell_{\max})} \cdot (1 - o(1)), \tag{9}$$

$$\mathbb{E}\left[T_i^w(\infty)\right]^{\text{FCFS}} \leq \frac{\sigma^2}{n(\delta - \ell_{\max})} \cdot (1 + o(1)). \tag{10}$$

*Consequently,*

$$\mathbb{E}\left[T^w(\infty)\right]^{\text{FCFS}} = \Theta\left( \frac{\sigma^2}{n\delta} \right), \quad \mathbb{E}\left[T_i^w(\infty)\right]^{\text{FCFS}} = \Theta\left( \frac{\sigma^2}{n\delta} \right). \tag{11}$$

Theorem 2 (Mean waiting time lower bound). *Consider the multiserver-job system with n servers satisfying Assumptions 1 and 2. Under any policy, the mean waiting time is lower bounded as*

$$\begin{aligned} \mathbb{E}\left[T^w(\infty)\right] &\geq \max_{i^* \leq i \leq I} \frac{1}{\lambda} \frac{\mu_{\min} \sigma_i^2}{\ell_i \delta_i} \cdot (1 - o(1)) \\ &= \Omega\left( \max_{i^* \leq i \leq I} \frac{1}{\lambda} \frac{\sigma_i^2}{\ell_i \delta_i} \right), \end{aligned} \tag{12}$$

*where $i^*$ is the critical index defined in* (8) *and $\mu_{\min} = \min_{i \in [I]} \mu_i$, and the expression represented by $o(1)$ is independent of the policies.*

Theorem 3 (Mean waiting time under P-Priority). *Consider the multiserver-job system with n servers satisfying Assumptions 1, 2, and 3. Under the P-Priority policy, the mean waiting time satisfies*

$$\begin{aligned} \mathbb{E}\left[T^w(\infty)\right]^{\text{P-Priority}} &\leq \frac{1}{\lambda} \sum_{i=i^*}^{I} \frac{\mu_{\max} \sigma_i^2}{\ell_i (\delta_i - \ell_i)} \cdot (1 + o(1)) \\ &= O\left( \max_{i^* \leq i \leq I} \frac{1}{\lambda} \frac{\sigma_i^2}{\ell_i \delta_i} \right), \end{aligned} \tag{13}$$

*where $i^*$ is the critical index defined in* (8) *and $\mu_{\max} = \max_{i \in [I]} \mu_i$. Consequently, the P-Priority policy achieves the optimal order of the mean waiting time.*

We have a more general bound for P-Priority policy that holds without Assumption 3. Interested readers can refer to the appendices of our technical report [18].

Below we state the results appearing in Section 1 as direct consequences to the above theorems.

Corollary 1 (Mean waiting times in the parameterized scaling regimes). *Consider the multiserver-job system with n servers satisfying Assumptions 1, 2 and 3. Suppose the maximal server need $\ell_{\max} = n^{\gamma}$ and the slack capacity $\delta = n^{\alpha}$, then the assumptions simplify to $0 \leq \gamma < \alpha < \frac{1+\gamma}{2} < 1$, $\rho_I = \Theta(1)$. We further assume that the total arrival rate $\lambda = \Theta(n)$. Then we have the following results:*

(a) *Under the FCFS policy, for each $i \in [I]$, the expected waiting time of type i jobs satisfies*

$$\mathbb{E}\left[T_i^w(\infty)\right]^{\text{FCFS}} = \Theta\left(n^{\gamma - \alpha}\right), \tag{14}$$

*and the mean waiting time over all job types also satisfies*

$$\mathbb{E}\left[T^w(\infty)\right]^{\text{FCFS}} = \Theta\left(n^{\gamma - \alpha}\right). \tag{15}$$

(b) *Under any policy, the mean waiting time is lower bounded as*

$$\mathbb{E}\left[T^w(\infty)\right] = \Omega\left(n^{-\alpha}\right), \tag{16}$$

*where the expression represented by $\Omega(n^{-\alpha})$ is independent of the policies.*

(c) *The mean waiting time under the P-Priority policy satisfies*

$$\mathbb{E}\left[T^w(\infty)\right]^{\text{P-Priority}} = \Theta\left(n^{-\alpha}\right). \tag{17}$$

## 4 PROOF ROADMAP AND DRIFT METHOD PRELIMINARIES

We organize our proofs of the main results as follows: we first prove two important bounds for a quantity called *workload* given by $\sum_{i=1}^{I} \frac{\ell_i}{\mu_i} Q_i$, in Lemma 1 and Lemma 2, respectively. Then we convert the workload bounds to the waiting time bounds in Theorem 1 and Theorem 2 using properties of FCFS and a linear programming relaxation. For Theorem 3, we analyze P-Priority by considering each $i$-th subsystems for $i \in [I]$. Some subsystems only need Lemma 1 and 2, while others require an additional Lemma 6.

Our proof approach is closely related to the recently developed drift method [12, 26]. The drift method allows us to extract information from a continuous-time Markov chain $\{S(t)\}_{t \geq 0}$ in state space $\mathcal{S}$ by computing the *drift* of different test functions. Specifically, let $f : \mathcal{S} \to \mathbb{R}$ be any function from the state space $\mathcal{S}$ to real numbers.

Because $S(t)$ is a Markov chain with countable state space and bounded transition rates, we can define *drift* of the function $f$ as

$$Gf(s) \triangleq \lim_{t\to 0} \mathbb{E}\left[\frac{f(S(t)) - f(s)}{t} \,\middle|\, S(0) = s\right]. \tag{18}$$

We call the operator $G$ the *generator* of the Markov chain.

For a multiserver-job system, let $I$-dimensional real vectors $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^I$ be possible realizations of state descriptors $\boldsymbol{X}(t)$ and $\boldsymbol{Z}(t)$, where recall that $\boldsymbol{X}(t)$ is the vector of the number of jobs in the system at time $t$, and $\boldsymbol{Z}(t)$ is the vector of the number of jobs in service at time $t$. We focus on $f$ that only depends on $\boldsymbol{x}$, i.e., $f : \mathbb{Z}_+^I \to \mathbb{R}$.

$$Gf(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{I} \lambda_i (f(\boldsymbol{x} + \boldsymbol{e}_i) - f(\boldsymbol{x})) + \sum_{i=1}^{I} \mu_i z_i (f(\boldsymbol{x} - \boldsymbol{e}_i) - f(\boldsymbol{x})), \tag{19}$$

where $\boldsymbol{e}_i \in \mathbb{R}^I$ is the vector whose $i$-th entry is 1 and all other entries are 0. Note that although $f$ and $Gf$ are functions of the system state $\boldsymbol{u}$, we write $f(\boldsymbol{x})$ and $Gf(\boldsymbol{x}, \boldsymbol{z})$ to highlight the variables that affect their values.

We frequently use the following relation regarding the drift

$$\mathbb{E}[Gf(\boldsymbol{X}, \boldsymbol{Z})] = 0. \tag{20}$$

Heuristically, this is because when $\boldsymbol{X}(0)$ and $\boldsymbol{Z}(0)$ follow the stationary distribution, $\boldsymbol{X}(t)$ and $\boldsymbol{Z}(t)$ also follow the stationary distribution, so $f(\boldsymbol{X}(t))$ and $f(\boldsymbol{X}(0))$ have the same expectation. Rigorously speaking, this relation only holds for well behaved functions and Markov processes. The conditions under which the relation holds are discussed in detail in our technical report [18]. Throughout the paper, we assume (20) holds for all $f$ that we consider.

## 5 WORKLOAD BOUNDS

In this section, we prove two bounds for a quantity called *workload* given by $\sum_{i=1}^{I} \frac{\ell_i}{\mu_i} Q_i$. These bounds are fundamental to the proofs of the main theorems. In Lemma 1, we give a lower bound on the expected workload applicable to *any policy*. In Lemma 2, we give upper bounds on the expected workload under any *$\delta'$-work-conserving policy*, a class of policies defined in Definition 1. The proof sketches of Lemma 1 and Lemma 2 are given in this section, and the complete proofs are provided in the appendices of [18].

Lemma 1 (Workload lower bound). *Consider the multiserver-job system with n servers satisfying Assumptions 1 and 2. Under any policy, the expected workload is lower bounded as*

$$\mathbb{E}\left[\sum_{i=1}^{I} \frac{\ell_i}{\mu_i} Q_i(\infty)\right] \geq \frac{\sigma^2}{\delta} \cdot (1 - o(1)), \tag{21}$$

*where the expression represented by $o(1)$ is independent of the policies.*

Definition 1. *We call a policy $\delta'$-work-conserving, if the following equation holds*

$$\sum_{i=1}^{I} \ell_i Z_i(t) \geq \min\left(\sum_{i=1}^{I} \ell_i X_i(t), n - \delta'\right) \quad \forall t \geq 0. \tag{22}$$

Here $\sum_{i=1}^{I} \ell_i Z_i(t)$ is equal to the number of busy servers at time $t$, while $\sum_{i=1}^{I} \ell_i X_i(t)$, which we call the *total server need*, is the potential number of busy servers if we can put all jobs at time $t$ into service. Therefore, under a $\delta'$-work-conserving policy, either all jobs are in service, or there are at most $\delta'$ idling servers. Under any $\ell_{\max}$-work-conserving policy, one can show that the system is stable when $\ell_{\max} \leq \epsilon_0 \delta$ (Assumption 2) holds. In particular, the system is stable under both FCFS and P-Priority.

Lemma 2 (Workload upper bound). *Consider the multiserver-job system with n servers under a $\delta'$-work-conserving policy with $\delta' \leq \epsilon_0 \delta$, where $\epsilon_0 \in (0, 1)$ is the parameter in Assumption 2. Then when $\delta = o\left(\frac{\sqrt{\sigma^2}}{\log n}\right)$,*

$$\mathbb{E}\left[\sum_{i=1}^{I} \frac{\ell_i}{\mu_i} Q_i(\infty)\right] \leq \frac{\sigma^2}{\delta - \delta'} \cdot (1 + o(1)) = O\left(\frac{\sigma^2}{\delta}\right); \tag{23}$$

*when $\delta = \Omega\left(\frac{\sqrt{\sigma^2}}{\log n}\right)$,*

$$\mathbb{E}\left[\sum_{i=1}^{I} \frac{\ell_i}{\mu_i} Q_i(\infty)\right] = O\left(\sqrt{\sigma^2} \log n\right). \tag{24}$$

*Remark.* When Assumption 1 is satisfied, i.e., when $\delta = o\left(\frac{\sqrt{\sigma^2}}{\log n}\right)$, the workload upper bound in Lemma 2 coincides with the workload lower bound in Lemma 1 order-wise, which implies that the expected workload $\mathbb{E}\left[\sum_{i=1}^{I} \frac{\ell_i}{\mu_i} Q_i(\infty)\right] = \Theta\left(\frac{\sigma^2}{\delta}\right)$. Note that in this case, although the expected workload under all $\delta'$-work-conserving policies has the same order, the mean waiting time can vary among policies, as shown for FCFS and P-Priority in Theorems 1 and 3.

***Preliminaries for proving Lemma 1 and Lemma 2.*** Our proofs focus on bounding the *normalized work*, defined as

$$\overline{W} \triangleq \sum_{i=1}^{I} \frac{\ell_i}{\mu_i} (X_i - \bar{x}_i),$$

where we write $\bar{x}_i \triangleq \frac{\lambda_i}{\mu_i}$ for notational simplicity. We claim that normalized work has the same expectation as the workload, i.e., $\mathbb{E}[\overline{W}] = \mathbb{E}\left[\sum_{i=1}^{I} \frac{\ell_i}{\mu_i} Q_i\right]$. To see this, recall that $Q_i = X_i - Z_i$. Now consider the drift of $X_i$, given by $GX_i = \lambda_i - \mu_i Z_i$. One can verify that $X_i$ satisfies $\mathbb{E}[GX_i] = 0$, and thus $\mathbb{E}[Z_i] = \frac{\lambda_i}{\mu_i}$. Therefore, the expected workload can be written as:

$$\mathbb{E}\left[\sum_{i=1}^{I} \frac{\ell_i}{\mu_i} Q_i\right] = \mathbb{E}\left[\sum_{i=1}^{I} \frac{\ell_i}{\mu_i} (X_i - Z_i)\right] = \mathbb{E}\left[\sum_{i=1}^{I} \frac{\ell_i}{\mu_i} (X_i - \bar{x}_i)\right] = \mathbb{E}[\overline{W}]. \tag{25}$$

Therefore, bounding the expected workload is equivalent to bounding the steady-state expectation of the normalized work $\mathbb{E}[\overline{W}]$.

We break $\mathbb{E}[\overline{W}]$ into three terms:

$$\mathbb{E}[\overline{W}] = \bar{r} + \mathbb{E}[(\overline{W} - \bar{r})^+] - \mathbb{E}[(\overline{W} - \bar{r})^-], \tag{26}$$

where $\bar{r} \in \mathbb{R}$ is up to our choice; $(\overline{W} - \bar{r})^+ \triangleq \max\{\overline{W} - \bar{r}, 0\}$ denotes the positive part, and $(\overline{W} - \bar{r})^- \triangleq -\min\{\overline{W} - \bar{r}, 0\}$ denotes the negative part.

The major difficulty during the proofs is bounding the expectation of the positive part $\mathbb{E}[(\overline{W} - \bar{r})^+]$. This relies on the relation $\mathbb{E}[Gf(\boldsymbol{X}, \boldsymbol{Z})] = 0$ introduced in Section 4. In our proofs, we choose $f$ to be piecewise quadratic functions to get bounds on the term

$$\mathbb{E}\left[\sum_{i=1}^{I} \ell_i (Z_i - \bar{x}_i) \cdot (\overline{W} - \bar{r})^+\right].$$

Since $\sum_{i=1}^{I} \ell_i(Z_i - \bar{x}_i) = \sum_{i=1}^{I} \ell_i Z_i - n + \delta$, we will be able to bound $\mathbb{E}[(\overline{W} - \bar{r})^+]$ if we are able to give an accurate estimate of the number of busy servers $\sum_{i=1}^{I} \ell_i Z_i$ when the normalized work $\overline{W} \geq \bar{r}$. To get a precise estimate, we exploit the state-space concentration result that says for each $i \in [I]$, $X_i$ cannot be much smaller than $\bar{x}_i$, i.e., $(X_i - \bar{x}_i)^-$ is small with high probability. Formally, this state-space concentration is established by Lemma 3, whose proof uses a sample-path coupling argument and is given in [18].

Lemma 3. *Consider the multiserver-job system with n servers. For any nonnegative vector* $c = (c_1, \ldots, c_I) \in \mathbb{R}_+^I$ *independent of* $n$*, let* $c_{\max} = \max_{i \in [I]} c_i$, $\mu_{\max} = \max_{i \in [I]} \mu_i$ *and let*

$$\Phi = \sum_{i=1}^{I} c_i \ell_i (X_i(\infty) - \bar{x}_i),$$

*where* $\bar{x}_i = \frac{\lambda_i}{\mu_i}$*. Then we have the three bounds below.*

*(a) For any* $K \geq 0$*,*

$$\mathbb{P}(\Phi \leq -K) \leq \exp\left(-\frac{K^2}{2c_{\max}^2 \mu_{\max} \sigma^2}\right). \tag{27}$$

*(b) For any* $\alpha \geq 0$ *and* $\beta \geq 0$ *such that* $\alpha\beta \geq c_{\max}^2 \mu_{\max} \sigma^2$ *and any* $j \geq 0$*,*

$$\mathbb{P}(\Phi \leq -\alpha - \beta j) \leq e^{-j}. \tag{28}$$

*(c) Let* $\Phi^- = \max\{-\Phi, 0\}$ *to be the negative part of* $\Phi$*. Then*

$$\mathbb{E}[\Phi^-] \leq \sqrt{c_{\max}^2 \mu_{\max} \sigma^2}. \tag{29}$$

Next, we give the proof sketches of Lemma 1 and Lemma 2.

***Proof sketch of Lemma 1 (workload lower bound).*** Recall that $\mathbb{E}[\overline{W}] = \bar{r}_1 + \mathbb{E}[(\overline{W} - \bar{r}_1)^+] - \mathbb{E}[(\overline{W} - \bar{r}_1)^-]$, for some scalar $\bar{r}_1$ to be specified later. To bound the positive part, we invoke the relation $\mathbb{E}[Gf(\boldsymbol{X}, \boldsymbol{Z})] = 0$ for a carefully constructed function $f(x)$ and get

$$\mathbb{E}\left[\sum_{i=1}^{I} \ell_i (Z_i - \bar{x}_i) \cdot (\overline{W} - \bar{r}_1)^+\right] \geq \sigma^2 - O(n\ell_{\max})\mathbb{P}\left(\overline{W} \leq \bar{r}_1 + \frac{\ell_{\max}}{\mu_{\min}}\right). \tag{30}$$

According to Lemma 3 (a) with $\Phi = \overline{W}$, we can choose some $\bar{r}_1 = -O\left(\sqrt{\sigma^2}\log n + \ell_{\max}\right)$ such that the probability on the right hand side is bounded by $\frac{1}{n^2}$. Moreover, observe that $\sum_{i=1}^{I} \ell_i(Z_i - \bar{x}_i) \leq n - (n - \delta) = \delta$. Therefore,

$$\mathbb{E}[(\overline{W} - \bar{r}_1)^+] \geq \frac{\sigma^2}{\delta} \cdot (1 - o(1)). \tag{31}$$

By Lemma 3 (c), we can immediately get that because $\bar{r}_1 \leq 0$, the negative part satisfies

$$\mathbb{E}[(\overline{W} - \bar{r}_1)^-] \leq \mathbb{E}[(\overline{W})^-] = O\left(\sqrt{\sigma^2}\right).$$

Combining the bounds on $\mathbb{E}\left[(\overline{W} - \bar{r}_1)^+\right]$ and $\mathbb{E}[(\overline{W} - \bar{r}_1)^-]$ gives

$$\mathbb{E}[\overline{W}] = \frac{\sigma^2}{\delta} \cdot (1 - o(1)) - O\left(\sqrt{\sigma^2}\log n + \ell_{\max}\right) = \frac{\sigma^2}{\delta} \cdot (1 - o(1)),$$

where the last equality follows from Assumption 1 and 2, that is, $\delta = o\left(\sqrt{\sigma^2}/\log n\right)$ and $\ell_{\max} \leq \epsilon_0 \delta$.

***Proof sketch of Lemma 2 (workload upper bound).*** Observe that $E[\overline{W}] \leq \bar{r}_2 + \mathbb{E}[(\overline{W} - \bar{r}_2)^+]$, for some $\bar{r}_2$ to be specified later. To bound the positive part $\mathbb{E}[(\overline{W} - \bar{r}_2)^+]$, we apply the relation $\mathbb{E}[Gf(\boldsymbol{X}, \boldsymbol{Z})] = 0$ to a carefully constructed function $f(x)$ and get

$$\mathbb{E}\left[\sum_{i=1}^{I} \ell_i (Z_i - \bar{x}_i) \cdot (\overline{W} - \bar{r}_2)^+\right] \leq \sigma^2. \tag{32}$$

In addition, we claim that there exists some $\gamma > 0$ such that

$$\gamma \cdot \mathbb{E}[(\overline{W} - \bar{r}_2)^+] \leq \mathbb{E}\left[\sum_{i=1}^{I} \ell_i (Z_i - \bar{x}_i) \cdot (\overline{W} - \bar{r}_2)^+\right] + o(1). \tag{33}$$

To prove this, we observe that both of the terms $\mathbb{E}[(\overline{W} - \bar{r}_2)^+]$ and $\mathbb{E}\left[\sum_{i=1}^{I} \ell_i (Z_i - \bar{x}_i) \cdot (\overline{W} - \bar{r}_2)^+\right]$ are non-zero only when

$$\sum_{i=1}^{I} \frac{\ell_i}{\mu_i}(X_i - \bar{x}_i) \geq \bar{r}_2. \tag{34}$$

By Lemma 3 (b) with $c_i = \frac{1}{\mu_{\min}} - \frac{1}{\mu_i}$, we also have the following inequality with probability at least $1 - \frac{1}{n^3}$,

$$\sum_{i=1}^{I} \left(\frac{1}{\mu_{\min}} - \frac{1}{\mu_i}\right) \ell_i (X_i - \bar{x}_i) \geq -K_2. \tag{35}$$

for some $K_2 = O\left(\sqrt{\sigma^2}\log n\right)$. Adding up the two inequalities above and applying $\delta'$-work-conserving property, we get

$$\sum_{i=1}^{I} \ell_i (Z_i - \bar{x}_i) \geq \min\left(\mu_{\min}(\bar{r}_2 - K_2), \delta - \delta'\right).$$

After handling the low probability event that (35) does not hold, we can show (33) with $\gamma = \min(\mu_{\min}(\bar{r}_2 - K_2), \delta - \delta')$. Therefore,

$$\begin{aligned} \mathbb{E}[\overline{W}] &\leq \bar{r}_2 + \mathbb{E}[(\overline{W} - \bar{r}_2)^+] \\ &\leq \bar{r}_2 + \frac{\sigma^2}{\min(\mu_{\min}(\bar{r}_2 - K_2), \delta - \delta')} + o(1). \end{aligned}$$

The upper bounds (23) and (24) in Lemma 2 follow once we choose a suitable $\bar{r}_2$. When $\delta = o\left(\sqrt{\sigma^2}/\log n\right)$, choosing $\bar{r}_2 = K_2 + (\delta - \delta')/\mu_{\min}$ yields $\mathbb{E}[\overline{W}] \leq \frac{\sigma^2}{\delta - \delta'} \cdot (1 + o(1))$. When $\delta = \Omega\left(\sqrt{\sigma^2}/\log n\right)$, choosing $\bar{r}_2 = K_2 + O\left(\sqrt{\sigma^2}\log n\right)$ yields $\mathbb{E}[\overline{W}] \leq O\left(\sqrt{\sigma^2}\log n\right)$.

## 6 PROOF SKETCH OF THEOREM 1 (WAITING TIMES UNDER FCFS)

The full proof of Theorem 1 is presented in our technical report [18]. Here we give a proof sketch. The proof is based on the intuition that, under FCFS, the number of type $i$ jobs in the queue is approximately proportional to its arrival rate $\lambda_i$, i.e.

$$\mathbb{E}[Q_i] \approx \frac{\lambda_i}{\lambda}\mathbb{E}[Q_\Sigma], \tag{36}$$

where $Q_\Sigma \triangleq \sum_{i=1}^{I} Q_i$ is the total queue length. Therefore, we can easily convert from the expected workload to mean waiting time:

$$\sum_{i=1}^{I} \frac{\ell_i}{\mu_i}\mathbb{E}[Q_i] \approx \sum_{i=1}^{I} \frac{\lambda_i \ell_i}{\mu_i}\frac{1}{\lambda}\mathbb{E}[Q_\Sigma] = (n - \delta)\mathbb{E}\left[T^w\right], \tag{37}$$

where the second equality is due to Little's law and the fact that $\sum_{i=1}^{I}\frac{\lambda_i\ell_i}{\mu_i} = n-\delta$.

This intuition is formalized by considering a *Modified-FCFS* policy, under which $\mathbb{E}^{\text{Modified-FCFS}}[Q_i] = \frac{\lambda_i}{\lambda}\mathbb{E}^{\text{Modified-FCFS}}[Q_\Sigma]$. Using the Modified-FCFS, we then construct an upper bounding system and a lower bounding system for the original FCFS system, such that the queue lengths of the original system are sandwiched between the queue lengths of the two modified systems.

## 7 PROOF OF THEOREM 2 (MEAN WAITING TIME LOWER BOUND)

Proof. Recall that by Little's law, $\mathbb{E}\left[T^w\right] = \frac{1}{\lambda}\sum_{i=1}^{I}\mathbb{E}[Q_i]$. Therefore, it suffices to show a lower bound on the total queue length. We fix a policy in the original system. For any $i$ with $i^* \le i \le I$, we consider the $i$-th subsystem by ignoring all job types with index greater than $i$. In the $i$-th subsystem, it is always possible to achieve the same $\mathbb{E}[Q_j]$'s for $j \le i$ by imitating the service decisions taken by the original system. Therefore, we have $\sum_{j=1}^{i}\frac{\ell_j}{\mu_j}\mathbb{E}[Q_j] \ge \frac{\sigma_i^2}{\delta_i}\cdot(1-o(1))$, where the right hand side expression is the workload lower bound of the $i$-th subsystem according to Lemma 1. Then the expected waiting time $\mathbb{E}[T^w]$ is lower-bounded by the optimal value of the following linear programming problem:

$$\begin{aligned}\min_{\{q_j:\, j\in[I]\}} \quad & \frac{1}{\lambda}\sum_{j=1}^{I} q_j \\ \text{subject to} \quad & \sum_{j=1}^{i}\frac{\ell_j}{\mu_j}q_j \ge \frac{\sigma_i^2}{\delta_i}\cdot(1-o(1)) \quad i^* \le i \le I \\ & q_j \ge 0 \quad \forall j \in [I],\end{aligned}$$

where $q_j$ corresponds to $\mathbb{E}[Q_j]$.

$$\frac{1}{\lambda}\sum_{j=1}^{I} q_j \ge \frac{1}{\lambda}\frac{\mu_{\min}}{\ell_i}\sum_{j=1}^{i}\frac{\ell_j}{\mu_j}q_j \ge \frac{\mu_{\min}\sigma_i^2}{\lambda\ell_i\delta_i}\cdot(1-o(1)), \tag{38}$$

for any $i^* \le i \le I$, where in the first inequality we have used the fact that $\mu_j \ge \mu_{\min}$ and $\ell_j \le \ell_i$ for any $j \le i$. Note that when $q_j$ is zero for each $j$ with $j \ne i$, the first inequality becomes an equality up to constant order factors in terms of $\mu_i$ and $\mu_{\min}$. Because the choice of $i$ with $i^* \le i \le I$ is arbitrary, we have that the optimal value is no less than

$$\max_{i^* \le i \le I}\frac{\mu_{\min}\sigma_i^2}{\lambda\ell_i\delta_i}\cdot(1-o(1)). \tag{39}$$

Therefore, $\mathbb{E}\left[T^w\right] \ge \max_{i^*\le i\le I}\frac{\mu_{\min}\sigma_i^2}{\lambda\ell_i\delta_i}\cdot(1-o(1))$. This completes the proof. □

*Remark.* The proof of the lower bound provides some intuitions for choosing the P-Priority policy. We consider the simple case where $i^* = I$. By (38), the total queue length orderwise achieves the lower bound when the queue consists of jobs with the largest server needs, which suggests us to give low priorities to those jobs. This is in a similar spirit to SRPT, which leaves jobs with large remaining service times in the queue (See, e.g., [16]).

## 8 PROOF OF THEOREM 3 (MEAN WAITING TIME UNDER P-PRIORITY)

To understand the behavior under P-Priority policy, one key observation is that for each $i \in [I]$, the type $i$ jobs are unaffected by type $j$ jobs with $j > i$. As a result, we can learn about the original system by analyzing each $i$-th subsystem, which is obtained by removing all jobs of type $j$ with $j > i$. Some subsystems are under relatively heavier traffic, or more precisely, subject to $\delta_i = o\left(\frac{\sqrt{\sigma_i^2}}{\log n}\right)$, while some subsystems are under lighter traffic. For those subsystems under relatively heavier traffic, Lemma 1 and Lemma 2 are enough for use; for those subsystems under lighter traffic, we sometimes use Lemma 6, which is a more refined bound on the expected total server need $\mathbb{E}\left[\sum_{i=1}^{I}\ell_iX_i\right]$, proved based on the two tail bounds in Lemma 4 and Lemma 5. The proofs of these Lemmas are in the appendices of [18].

Lemma 4. *Consider the multiserver-job system with n servers satisfying $\ell_{\max} \le \epsilon_0\delta$ (Assumption 2). Letting $\bar{x}_i = \frac{\lambda_i}{\mu_i}$, under any $\ell_{\max}$-work-conserving policy, the normalized work has the following tail bound: for any $\epsilon$ such that $0 < \epsilon < \epsilon_0$, there exists $\alpha_1 = \frac{2\epsilon\delta}{\mu_{\min}} + \Theta\left(\frac{n\ell_{\max}}{\delta}\log n\right)$ and $\beta_1 = \Theta\left(\frac{n\ell_{\max}}{\delta}\right)$ such that for any $j \ge 0$,*

$$\mathbb{P}\left(\sum_{i=1}^{I}\frac{\ell_i}{\mu_i}(X_i(\infty)-\bar{x}_i) \ge \alpha_1 + \beta_1\cdot j\right) \le e^{-j}. \tag{40}$$

Lemma 5. *Consider the multiserver-job system with n servers satisfying $\ell_{\max} \le \epsilon_0\delta$ (Assumption 2). Letting $\bar{x}_i = \frac{\lambda_i}{\mu_i}$, under any $\ell_{\max}$-work-conserving policy, the total server need has the following tail bound: there exists $\alpha_2$ and $\beta_2$ with $\alpha_2 = \frac{\delta}{2} + \Theta(\frac{n\ell_{\max}}{\delta}\log n)$ and $\beta_2 = \Theta(\frac{n\ell_{\max}}{\delta})$ such that for any $j \ge 0$,*

$$\mathbb{P}\left(\sum_{i=1}^{I}\ell_i(X_i(\infty)-\bar{x}_i) \ge \alpha_2 + \beta_2\cdot j\right) \le e^{-j}. \tag{41}$$

The proof technique of the two lemmas is using state space concentration successively: Lemma 4 relies on the state-space concentration implied by Lemma 3, while Lemma 5 relies on the state-space concentration implied by Lemma 3 and Lemma 4.

As a consequence of the first two lemmas, we can give a bound on the expectation of the total server need $\mathbb{E}[\sum_{i=1}^{I}\ell_iX_i]$, in a different traffic regime than what is assumed in Assumption 1. This is useful for analyzing the dynamics of subsystems under P-Priority.

Lemma 6 (Total server need upper bound under a lighter traffic). *Consider the multiserver-job system with n servers satisfying $\ell_{\max} \le \epsilon_0\delta$ (Assumption 2), and $\delta = \omega\left(\sqrt{n\ell_{\max}}\log n\right)$. Letting $\bar{x}_i = \frac{\lambda_i}{\mu_i}$, under any $\ell_{\max}$-work-conserving policy, the expected total server need has the following upper bound:*

$$\mathbb{E}\left[\sum_{i=1}^{I}\ell_i(X_i(\infty)-\bar{x}_i)\right] = \exp\left(-\Omega\left(\frac{\delta^2}{n\ell_{\max}}\right)\right). \tag{42}$$

As a quick digression, with Lemma 5, we can prove an upper bound on the queueing probability in Corollary 2. We comment that this queueing probability bound significantly improves on the bound in [36] in a slightly more constrained traffic regime.

Corollary 2 (Queueing probability). *Consider the multiserver-job system with $n$ servers satisfying $\ell_{\max} \le \epsilon_0 \delta$ (Assumption 2), and $\delta = \omega\left(\sqrt{n\ell_{\max}} \log n\right)$. Then the probability that a job arrival in steady-state experiencing queueing has the following upper bound:*

$$\mathbb{P}\left(\sum_{i=1}^{I} \ell_i X_i(\infty) \ge n\right) = \exp\left(-\Omega\left(\frac{\delta^2}{n\ell_{\max}}\right)\right). \tag{43}$$

Now we are ready to prove Theorem 3.

Proof of Theorem 3. Recall that by Little's Law, we have that $\mathbb{E}[T^w] = \frac{1}{\lambda}\sum_{i=1}^{I}\mathbb{E}[Q_i]$, and thus it suffices to bound $\mathbb{E}[Q_i]$'s. We bound $\mathbb{E}[Q_i]$ separately for each $i \in [I]$ as follows:

$$\mathbb{E}[Q_i] \le \frac{1}{\ell_i}\mathbb{E}\left[\sum_{j=1}^{i} \ell_j Q_j\right] \le \frac{1}{\ell_i}\mathbb{E}\left[\sum_{j=1}^{i} \ell_j (X_j - \bar{x}_j)\right], \tag{44}$$

where we have used the fact that $Q_i$'s are non-negative, $Q_j = X_j - Z_j$ and $\mathbb{E}[Z_j] = \frac{\lambda_j}{\mu_j} = \bar{x}_j$. Observe that under P-Priority, the dynamics of the first $i$ types of jobs are unaffected by the rest of the jobs. Therefore, the expectation $\mathbb{E}\left[\sum_{j=1}^{i} \ell_j (X_j - \bar{x}_j)\right]$ can be viewed as the expected total server need in the $i$-th subsystem, which has slack capacity $\delta_i$, maximal server need $\ell_i$, and work variability $\sigma_i^2$. We discuss the bound on $\mathbb{E}[Q_i]$ in three cases based on different relationships of $\delta_i$, $\ell_i$ and $\sigma_i^2$.

**Case 1:** $\delta_i = o\left(\sqrt{\frac{\sigma_i^2}{\log n}}\right)$. Applying Lemma 2 to the $i$-th subsystem, we have

$$\mathbb{E}\left[\sum_{j=1}^{i} \ell_j (X_j - \bar{x}_j)\right] \le \mu_{\max}\mathbb{E}\left[\sum_{j=1}^{i} \frac{\ell_j}{\mu_j}(X_j - \bar{x}_j)\right] \le \frac{\mu_{\max}\sigma_i^2}{\delta_i - \ell_i}\cdot(1 + o(1)).$$

Therefore, $\mathbb{E}[Q_i] \le \frac{\mu_{\max}\sigma_i^2}{\ell_i(\delta_i - \ell_i)}\cdot(1 + o(1))$.

**Case 2:** $\delta_i = \Omega\left(\sqrt{\frac{\sigma_i^2}{\log n}}\right)$ **and** $\ell_i = \omega\left(\frac{\delta \log n}{\sqrt{\sigma^2}}\ell_{\max}\right)$. Applying Lemma 2 to the $i$-th subsystem, we have

$$\mathbb{E}\left[\sum_{j=1}^{i} \ell_j (X_j - \bar{x}_j)\right] \le \mu_{\max}\mathbb{E}\left[\sum_{j=1}^{i} \frac{\ell_j}{\mu_j}(X_j - \bar{x}_j)\right] = O\left(\sqrt{\sigma_i^2}\log n\right).$$

Therefore, $\mathbb{E}[Q_i] \le O\left(\frac{\sqrt{\sigma_i^2}\log n}{\ell_i}\right) = o\left(\frac{\sigma^2}{\delta}\cdot\frac{1}{\ell_{\max}}\right)$, where the equality is due to $\ell_i = \omega\left(\frac{\delta \log n}{\sqrt{\sigma^2}}\ell_{\max}\right)$ and $\sigma_i^2 \le \sigma^2$.

**Case 3:** $\delta_i = \Omega\left(\sqrt{\frac{\sigma_i^2}{\log n}}\right)$ **and** $\ell_i = O\left(\frac{\delta \log n}{\sqrt{\sigma^2}}\ell_{\max}\right)$. We first show that $\delta_i = \omega\left(\sqrt{n\ell_i}\log n\right)$. To see this, observe that $\delta_i = \Omega\left(\sqrt{\frac{\sigma_i^2}{\log n}}\right)$ implies $i < I$, so $\delta_i \ge \frac{\lambda_I \ell_I}{\mu_I} = n\rho_I$. By Assumption 3, we have

$$\delta_i \ge n\rho_I = \omega\left(\sqrt{\frac{\delta \log n}{\sqrt{\sigma^2}}\cdot n\ell_{\max}\log n}\right) = \omega\left(\sqrt{n\ell_i}\log n\right). \tag{45}$$

Therefore, we can apply Lemma 6 to the $i$-th subsystem to get:

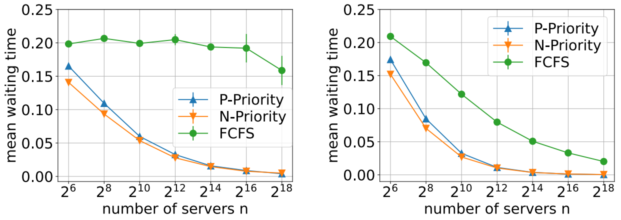
$$\mathbb{E}\left[\sum_{j=1}^{i} \ell_j (X_j - \bar{x}_j)\right] = \exp\left(-\Omega\left(\frac{\delta_i^2}{n\ell_i}\right)\right) \le \exp\left(-\Omega\left((\log n)^2\right)\right),$$

where we have used $\delta_i = \omega\left(\sqrt{n\ell_i}\log n\right)$ in the inequality. Therefore, $\mathbb{E}[Q_i] \le \exp\left(-\Omega\left((\log n)^2\right)\right)$, decaying faster than any polynomial.

Combining the three cases, we get

$$\mathbb{E}\left[T^w\right] \le \frac{1}{\lambda}\sum_{i=i^*}^{I}\frac{\mu_{\max}\sigma_i^2}{\ell_i(\delta_i - \ell_i)}\cdot(1 + o(1)),$$

where we have used the fact that $\mathbb{E}[T^w] = \frac{1}{\lambda}\sum_{i=1}^{I}\mathbb{E}[Q_i]$. The summation is taken from $i^*$ to $I$ because Case 1 corresponds to $i^* \le i \le I$, and the $E[Q_i]$ of another two cases are of lower orders. Finally, because $I$ and $\mu_{\max}$ are independent of $n$, and $\ell_i \le \ell_{\max} \le \epsilon_0\delta \le \epsilon_0\delta_i$ for some $\epsilon_0 < 1$, we have $\mathbb{E}[T^w] = O\left(\max_{i^* \le i \le I}\frac{1}{\lambda}\frac{\sigma_i^2}{\ell_i\delta_i}\right)$. $\square$



(a) Mean waiting time under the Parameter Set One. There are three job types, with service rates and server needs given by $\mu_1 = 0.25$, $\ell_1 = 1$; $\mu_2 = 0.5$, $\ell_2 = \lfloor \log_2 n \rfloor$; $\mu_3 = 1$, $\ell_3 = \lfloor \sqrt{n} \rfloor$; slack capacity $\delta = 2\lfloor \sqrt{n} \rfloor$; arrival rates $\lambda_i$'s are chosen so that loads $\rho_1 = \rho_2 = \rho_3 = \frac{n-\delta}{3n}$.

(b) Mean waiting time under the Parameter Set Two. There are three job types, with the same service rates, server needs and slack capacity as the Parameter Set One. The loads are given by $\rho_1 = \rho_2 = \frac{n-\delta-n^{0.7}}{2n}$, $\rho_3 = n^{-0.3}$. The arrival rates $\lambda_i$'s are chosen according to loads, service rates and server needs.

**Figure 3: The mean waiting times of FCFS, P-Priority, and N-Priority under two sets of parameters.**

## 9 SIMULATION RESULTS

We perform simulation experiments to demonstrate the mean waiting time under FCFS, P-Priority and N-Priority, where N-Priority is the non-preemptive variant of P-Priority that serves a job with the smallest server need in the queue when enough servers free up.

We run the simulation experiment under two sets of parameters. The Parameter Set One satisfies all three assumptions, while the Parameter Set Two does not satisfy Assumption 3. The parameters are specified in the caption of Figure 3.

We plot the mean waiting time against the number of servers $n$ under the three policies, as shown in Figure 3. The parameter $n$ takes value in $\{2^6, 2^8, 2^{10}, 2^{12}, 2^{14}, 2^{16}, 2^{18}\}$. For each data point, we run a long trajectory to estimate the mean and the confidence interval. The confidence interval is estimated through batch means method [5], which divides the trajectory into 20 batches, and calculates the variance of the means of each batch. It turns out that the confidence intervals of most data points in the plots are too small to be visible.

We have the following observations from the experiments. First, there is a large performance gap between FCFS and P-Priority for systems with finite number of servers $n$, which complements our asymptotic results. Note that although in the Parameter Set Two, the absolute different between FCFS and P-Priority seems to close up as $n$ gets large, their ratio is always greater than 3 when $n \geq 2^{10}$, and it gets to as large as 40.0 under Parameter Set One, and 54.4 under Parameter Set Two. Second, N-Priority performs comparably with P-Priority, and sometimes even performs slightly better.

## 10 CONCLUSION AND FUTURE WORK

In this paper, we have established order-wise sharp bounds on the mean waiting times of multiserver jobs under FCFS and P-Priority. We have also proved a lower bound of mean waiting time applicable to any policy. Those bounds imply the optimality of P-Priority and the strict sub-optimality of FCFS. Apart from the theoretical analysis, we have also demonstrated through simulations the performance improvement of P-Priority compared with FCFS in finite systems, and the fact that N-Priority, which is the non-preemptive variant of P-Priority, has comparable performance with P-Priority.

There are several interesting directions for future work: (i) Derive a tighter bound on the mean waiting time under P-Priority when the commonness assumption is violated. (ii) Analyze the performance of N-Priority. (iii) Relax the maximal server need assumption (Assumption 2), which may require new policy designs to ensure stability while keeping the mean waiting time small.

## ACKNOWLEDGMENTS

This work was supported in part by NSF CNS Grant #200773 and NSF ECCS Grant #2145713.

## REFERENCES

[1] 2022. *Google Kubernetes Engine (GKE).* https://cloud.google.com/kubernetes-engine

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *Proc. USENIX Conf. Operating Systems Design and Implementation (OSDI).* USENIX Association, Savannah, GA, 265–283.

[3] Larisa Afanaseva, Elena Bashtova, and Svetlana Grishunina. 2019. Stability analysis of a multi-server model with simultaneous service and a regenerative input flow. *Methodol. Comp. Appl. Probab.* 22, 4 (2019), 1–17.

[4] E. Arthurs and J. Kaufman. 1979. Sizing a Message Store Subject to Blocking Criteria. In *Proc. Int. Symp. Computer Performance, Modeling, Measurements and Evaluation (IFIP Performance).* North-Holland Publishing Co., NLD, 547–564.

[5] Søren Asmussen and Peter W. Glynn. 2007. *Steady-State Simulation.* Springer New York, New York, NY, 96–125.

[6] Baris Ata and Itai Gurvich. 2012. On optimality gaps in the halfin-whitt regime. *Annals of Applied Probability* 22, 1 (2012), 407–455.

[7] Rami Atar. 2012. A Diffusion Regime with Nondegenerate Slowdown. *Operations Research* 60, 2 (2012), 490–500.

[8] Rami Atar, Avi Mandelbaum, and Martin I. Reiman. 2004. Scheduling a multi class queue with many exponential servers: Asymptotic optimality in heavy traffic. *Annals of Applied Probability* 14, 3 (2004), 1084–1134.

[9] N. G. Bean, R. J. Gibbens, and S. Zachary. 1995. Asymptotic Analysis of Single Resource Loss Systems in Heavy Traffic, with Applications to Integrated Networks. *Adv. Appl. Probab.* 27, 1 (March 1995), 273–292.

[10] Percy H. Brill and Linda Green. 1984. Queues in Which Customers Receive Simultaneous Service from a Random Number of Servers: A System Point Approach. *Manage. Sci.* 30, 1 (1984), 51–68.

[11] Andrew Daw and Jamol Pender. 2019. On the Distributions of Infinite Server Queues with Batch Arrivals. *Queueing Syst.* 91, 3–4 (apr 2019), 367–401.

[12] Atilla Eryilmaz and R. Srikant. 2012. Asymptotically Tight Steady-state Queue Length Bounds Implied by Drift Conditions. *Queueing Syst.* 72, 3-4 (Dec. 2012), 311–359.

[13] Dimitrios Filippopoulos and Helen Karatza. 2006. A Two-Class Parallel Queue with Pure Space Sharing among Rigid Jobs and General Service Times. In *Proc. EAI Int. Conf. Performance Evaluation Methodologies and Tools (VALUETOOLS).* Association for Computing Machinery, 2–es.

[14] Isaac Grosof, Mor Harchol-Balter, and Alan Scheller-Wolf. 2020. Stability for Two-class Multiserver-job Systems. *arXiv:2010.00631 [cs.PF]* (Oct. 2020).

[15] Shlomo Halfin and Ward Whitt. 1981. Heavy-Traffic Limits for Queues with Many Exponential Servers. *Oper. Res.* 29, 3 (1981), 567–588.

[16] Mor Harchol-Balter. 2013. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action* (1st ed.). Cambridge University Press, New York, NY.

[17] J. Michael Harrison and Assaf Zeevi. 2004. Dynamic scheduling of a multiclass queue in the halfin-whitt heavy traffic regime. *Oper. Res.* 52, 2 (2004), 1–31.

[18] Yige Hong and Weina Wang. 2021. *Sharp Waiting-Time Bounds for Multi-Server Jobs.* Technical Report. Carnegie Mellon University. https://arxiv.org/abs/2109.05343

[19] P. J. Hunt and C. N. Laws. 1997. Optimization via trunk reservation in single resource loss systems under heavy traffic. *Ann. Appl. Probab.* 7, 4 (Nov. 1997), 1058–1079.

[20] Sung-Han Lin, Marco Paolieri, Cheng-Fu Chou, and Leana Golubchik. 2018. A Model-Based Approach to Streamlining Distributed Training for Asynchronous SGD. In *IEEE Int. Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS).* 306–318.

[21] Xin Liu. 2019. *Steady State Analysis of Load Balancing Algorithms in the Heavy Traffic Regime.* Ph.D. Dissertation. Arizona State University.

[22] Xin Liu, Kang Gong, and Lei Ying. 2022. Steady-state analysis of load balancing with Coxian-2 distributed service times. *Naval Research Logistics (NRL)* 69, 1 (2022), 57–75.

[23] Xin Liu and Lei Ying. 2019. On Universal Scaling of Distributed Queues under Load Balancing. *arXiv:1912.11904 [math.PR]* (2019).

[24] Xin Liu and Lei Ying. 2020. Steady-state analysis of load-balancing algorithms in the sub-Halfin–Whitt regime. *J. Appl. Probab.* 57, 2 (2020), 578–596.

[25] Siva Theja Maguluri and R. Srikant. 2013. Scheduling jobs with unknown duration in clouds. In *Proc. IEEE Int. Conf. Computer Communications (INFOCOM).* 1887–1895.

[26] Siva Theja Maguluri and R. Srikant. 2016. Heavy traffic queue length behavior in a switch under the MaxWeight algorithm. *Stoch. Syst.* 6, 1 (2016), 211–250.

[27] Evsey Morozov and Alexander S. Rumyantsev. 2016. Stability Analysis of a MAP/M/s Cluster Model by Matrix-Analytic Method. In *European Workshop Computer Performance Engineering (EPEW)*, Vol. 9951. Chios, Greece, 63–76.

[28] 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.* Association for Computing Machinery, Irvine, CA, 67–69.

[29] Alexander Rumyantsev and Evsey Morozov. 2017. Stability criterion of a multi-server model with simultaneous service. *Ann. Oper. Res.* 252, 1 (2017), 29–39.

[30] Oleg M. Tikhonenko. 2005. Generalized Erlang Problem for Service Systems with Finite Total Capacity. *Probl. Inf. Transm.* 41, 3 (2005), 243–253.

[31] 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.

[32] Mark van der Boor, Martin Zubeldia, and Sem Borst. 2020. Zero-wait load balancing with sparse messaging. *Oper. Res. Lett.* 48, 3 (2020), 368–375.

[33] Nico M. van Dijk. 1989. Blocking of Finite Source Inputs Which Require Simultaneous Servers with General Think and Holding Times. *Oper. Res. Lett.* 8, 1 (Feb. 1989), 45 – 52.

[34] Abhishek Verma, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. 2015. Large-scale cluster management at Google with Borg. In *Proc. European Conf. Computer Systems (EuroSys).* Bordeaux, France, 18.

[35] Weina Wang, Siva Theja Maguluri, R. Srikant, and Lei Ying. 2018. Heavy-Traffic Delay Insensitivity in Connection-Level Models of Data Transfer with Proportionally Fair Bandwidth Sharing. *SIGMETRICS Perform. Eval. Rev.* 45, 3 (mar 2018), 232–245.

[36] Weina Wang, Qiaomin Xie, and Mor Harchol-Balter. 2021. Zero Queueing for Multi-Server Jobs. *Proc. ACM Meas. Anal. Comput. Syst.* 5, 1, Article 07 (Feb. 2021), 25 pages.

[37] Wentao Weng and Weina Wang. 2020. Achieving Zero Asymptotic Queueing Delay for Parallel Jobs. *Proc. ACM Meas. Anal. Comput. Syst.* 4, 3, Article 42 (Nov. 2020), 36 pages.

[38] Wentao Weng, Xingyu Zhou, and R. Srikant. 2020. Optimal Load Balancing with Locality Constraints. *Proc. ACM Meas. Anal. Comput. Syst.* 4, 3, Article 45 (nov 2020), 37 pages.

[39] Ward Whitt. 1985. Blocking when service is required from several facilities simultaneously. *AT&T Tech. J.* 64 (1985), 1807 – 1856.

[40] 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.

[41] Martin Zubeldia. 2020. Delay-Optimal Policies in Partial Fork-Join Systems with Redundancy and Random Slowdowns. *Proc. ACM SIGMETRICS Int. Conf. Measurement and Modeling of Computer Systems* 4, 1, Article 02 (May 2020), 49 pages.