Zero Queueing for Multi-Server Jobs

WEINA WANG, Carnegie Mellon University, USA
QIAOMIN XIE, Cornell University, USA
MOR HARCHOL-BALTER, Carnegie Mellon University, USA

Cloud computing today is dominated by *multi-server jobs*. These are jobs that request multiple servers *simultaneously* and hold onto all of these servers for the duration of the job. Multi-server jobs add a lot of complexity to the traditional one-server-per-job model: an arrival might not "fit" into the available servers and might have to queue, blocking later arrivals and leaving servers idle. From a queueing perspective, almost nothing is understood about multi-server job queueing systems; even understanding the exact stability region is a very hard problem.

In this paper, we investigate a multi-server job queueing model under scaling regimes where the number of servers in the system grows. Specifically, we consider a system with multiple classes of jobs, where jobs from different classes can request different numbers of servers and have different service time distributions, and jobs are served in first-come-first-served order. The multi-server job model opens up new scaling regimes where *both* the number of servers that a job needs and the system load scale with the total number of servers. Within these scaling regimes, we derive the first results on stability, queueing probability, and the transient analysis of the number of jobs in the system for each class. In particular we derive sufficient conditions for zero queueing. Our analysis introduces a novel way of extracting information from the Lyapunov drift, which can be applicable to a broader scope of problems in queueing systems.

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

ACM Reference Format:
Weina Wang, Qiaomin Xie, and Mor Harchol-Balter. 2021. Zero Queueing for Multi-Server Jobs. *Proc. ACM Meas. Anal. Comput. Syst.* 5, 1, Article 7 (March 2021), 25 pages. https://doi.org/10.1145/3447385

1 INTRODUCTION

Queueing theorists have long been interested in characterizing the probability that an arriving job will have to queue, i.e., the *queueing probability*. This question has a rich history where it has been investigated under different scaling regimes. Consider the classical M/M/n system that has load $1 - \beta n^{-\alpha}$ with $0 < \beta < 1$ and $\alpha \geq 0$. The case of $\alpha = 1/2$ is known as the Halfin-Whitt regime [17] and serves as the critical threshold that separates diminishing queueing probability from non-diminishing queueing probability, as the number of servers n grows. In particular, it is known that the queueing probability is diminishing when $0 \leq \alpha < 1/2$ (sub-Halfin-Whitt); is strictly between 0 and 1 when $\alpha = 1/2$ (Halfin-Whitt); and goes to 1 when $\alpha > 1/2$ (see, e.g., [9]).

Today's computing clusters are more general than the M/M/n model used in the past. In particular, the jobs typically request *multiple servers simultaneously* and hold onto them for the duration of

Authors' addresses: Weina Wang, weinaw@cs.cmu.edu, Carnegie Mellon University, Computer Science Department, 5000 Forbes Ave, Pittsburgh, PA, USA, 15213; Qiaomin Xie, qiaomin.xie@cornell.edu, Cornell University, School of Operations Research and Information Engineering, 237 Frank H.T. Rhodes Hall, Ithaca, NY, USA, 14853; Mor Harchol-Balter, harchol@cs.cmu.edu, Carnegie Mellon University, Computer Science Department, 5000 Forbes Ave, Pittsburgh, PA, USA, 15213.

2476-1249/2021/3-ART7. https://doi.org/10.1145/3447385

the job. This difference is largely a result of machine learning jobs like TensorFlow [1, 24], which are highly parallel. For example, when we look at Google's Borg Scheduler [48], we see that the number of servers occupied by an individual job can be anywhere from 1 to 100000, as illustrated in Figure 1 [46, 55]. We refer to jobs that occupy multiple servers/cores as *multi-server jobs*. While multi-server jobs have always existed in the niche supercomputing world, they have now become mainstream.

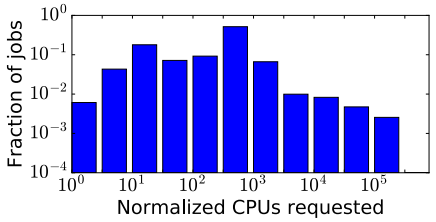

Fig. 1. Distribution of the number of CPU cores requested by individual Google jobs according to Google's recently published Borg Trace [46, 55] as taken from [15]. For privacy reasons, Google publishes only normalized numbers, however it is still clear that the range spans 5 orders of magnitude across different jobs.

The advent of multi-server jobs requires us to generalize our queueing models. Figure 2 shows an illustration of what we will refer to as *the multi-server job queueing model*. Here there are a total of n servers. Jobs arrive with average rate λ and are served in First-Come-First-Serve (FCFS) order [1]. With probability p_i an arrival is of class i. Each arriving job of class i requests m_i servers and holds onto these servers for time distributed according to random variable S_i. We will refer to the number of servers a job demands as the *server need* of the job, while we refer to the time that the job holds onto the servers as its *service time*.

The multi-server job queueing model opens up new scaling regimes where both the server needs of jobs and the system load scale with the number of servers. In this paper we ask:

Under this new joint scaling regime, when is diminishing queueing probability achievable?

The property of *diminishing queueing probability* is also referred to as (asymptotically) *zero queueing* in the literature, and we will use these two terms interchangeably. The answer to this question can take a very different form from the classical results. To see this, let us consider the following simple example. Suppose jobs arrive to a system with n servers according to a Poisson process of rate λ where all jobs are of the *same* class 1. Suppose that $S_1 \sim \text{Exp}(\mu_1)$ and $m_1 = n/2$ servers. Even in this degenerate version of the multi-server job model, the work associated with a job depends both on its service time, S_1, and also on its server need, m_1. Therefore, we redefine the notion of load to be $\rho \triangleq \frac{\lambda m_1 \mathbb{E}[S_i]}{n} = \frac{\lambda}{2\mu_1}$. Then even when the load ρ is a positive constant smaller than 1, analogous to the classical subcritical regime in a large system [20], the queueing probability does not diminish when n grows. The reason is that the large server need of jobs makes the system equivalent to an M/M/2 system, which is effectively a small system even though the actual number of servers grows.

Understanding the queueing probability becomes much more challenging when there are multiple classes of jobs that have heterogeneous server needs and service time distributions. Here it is often the case that some servers remain idle, because the job at the head of the queue cannot "fit" into

[1]Importantly, we note that we are assuming that jobs arrive to a centralized queue and are served in FCFS order, rather than trying to "pack" jobs into servers. A centralized FCFS scheduler is the default scheduler used in the cloud-computing industry when running multi-server jobs [46, 48]. Even when there are multiple priority classes of jobs, as in [46], within each class the jobs are served in FCFS order.

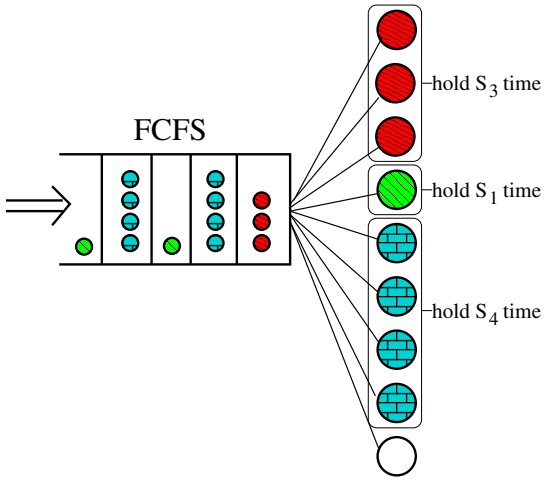

Fig. 2. The multi-server job queueing model with $n = 9$ servers. An arriving job of class i requests m_i servers and holds onto these servers for time distributed according to random variable S_i. In this particular illustration, $m_i = i$. Note that the job at the head of the queue cannot enter service because it does not fit; hence one server is idle.

the remaining unused servers, as illustrated in Figure 2. Although there may exist jobs in the queue with smaller server needs, those jobs are blocked by the head-of-queue job. Due to this *head-of-queue blocking*, the process that tracks the number of each class of jobs in service is highly complicated. Almost nothing is known on the performance of the multi-server job queueing model. Attempts to derive the steady-state distribution have assumed highly simplified systems with only $n = 2$ servers [10, 14], where solutions are already highly complex, involving roots to a quartic equation. Even characterizing the stability region of the system is an open problem except for the special cases where all jobs have the same service rates [2, 36, 41], or where there are only two job classes with different service rates [15].

Our results

We consider a system of n servers and K classes of jobs. For system parameters that are functions of n, we add a superscript $^{(n)}$ to indicate the dependency unless otherwise specified. Jobs arrive according to a Poisson process with rate $\lambda^{(n)}$ and an arrival is of class i with probability $p_i^{(n)}$. Then for each job class i with $1 \le i \le K$, the arrivals form a Poisson process with rate $\lambda_i^{(n)} := \lambda^{(n)} p_i^{(n)}$. Jobs of class i have i.i.d. service times exponentially distributed with rate μ_i, which does not scale with n. Each class i job has a server need of $m_i^{(n)}$ servers. Let $m_{\max}^{(n)} = \max_{1 \le i \le K} m_i^{(n)}$ denote the maximum server need.

Stability result. We first define a notion of *load* for multi-server job queueing systems, which will be used in the stability condition. Traditionally, for single-server jobs, the *work* brought in by a job is quantified by its service time, i.e., the time the job needs to occupy a server. However, for multi-server jobs, we need to account for the fact that they occupy multiple servers simultaneously. Therefore, we quantify the *work* brought in by a multi-server job using the product "server need $\times$ service time", which is consistent with the CPU-hours metric used in practice [46]. Then we define the load of class i jobs to be $\rho_i^{(n)} = \frac{\lambda_i^{(n)} m_i^{(n)}}{n\mu_i}$, and the total system load to be $\rho^{(n)} = \sum_{i=1}^{K} \rho_i^{(n)}$.

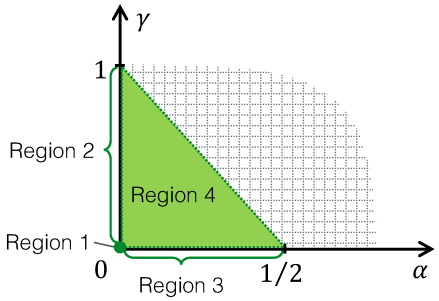

Fig. 3. Joint scaling regimes for system load $\rho^{(n)} = 1 - \beta n^{-\alpha}$ and server need $m_{\max}^{(n)} = \Theta(n^{\gamma})$.

We then show in Theorem 4.1 that a sufficient condition for the system to be stable is that

$$\rho^{(n)} < 1 - \frac{m_{\max}^{(n)}}{n}. \tag{1}$$

Note that when the load $\rho^{(n)} > 1$, no policy can stabilize the system. Therefore, the condition $\rho^{(n)} < 1 - \frac{m_{\max}^{(n)}}{n}$ is *asymptotically tight* when $m_{\max}^{(n)} = o(n)$. [2] We comment that this sufficient condition is in general not tight non-asymptotically, and finding the exact stability region is a hard open problem. We refer the readers to the section on related works (Section 2) for more details.

Queueing probability. The most important result in this paper is our characterization of the queueing probability, denoted by $P_Q^{(n)}$, which refers to the probability that a job cannot enter service immediately upon arrival in steady state. Specifically, we chart the joint scaling regimes for the server needs of jobs and the system load to answer the question of when *diminishing queueing probability* is achievable, i.e., when $P_Q^{(n)} \to 0$ as $n \to \infty$, for the multi-server job queueing model.

For ease of exposition, we parameterize the scaling regimes as the number of servers n grows in the following way. Assume that the load $\rho^{(n)} = 1 - \beta n^{-\alpha}$ with $0 < \beta < 1$ and $\alpha \geq 0$ and the maximum server need $m_{\max}^{(n)} = \Theta(n^{\gamma})$ with $0 \leq \gamma \leq 1$. We focus on the key parameters α and γ and divide the regimes into four regions as illustrated in Figure 3. We will start our discussion with results in simpler regions, Regions 1–3, since they are closely connected to classical queueing models. Then we will turn to results in the *main region of focus*, Region 4, which corresponds to the novel scaling regime where the load and the server needs scale *jointly*.

- **Region 1:** When the load $\rho^{(n)}$ stays *constant* ($\alpha = 0$) and the server needs also stay *constant* ($\gamma = 0$), does the queueing probability, $P_Q^{(n)}$, diminish as $n \to \infty$?

  Our result: We show that $P_Q^{(n)}$ is indeed diminishing in this region. This result should not be surprising given the classical result that when jobs have unit server need, the queueing probability is diminishing under constant load (see, e.g., [17], where a special case of their results analyzes a system where each job has a server need of one).

- **Region 2:** When the load $\rho^{(n)}$ stays *constant* ($\alpha = 0$) but the server needs *scale* ($\gamma > 0$), does the queueing probability, $P_Q^{(n)}$, diminish as $n \to \infty$?

  Our result: We show that $P_Q^{(n)}$ is diminishing when $\gamma < 1$, i.e., when a job drawn from the class of highest server need occupies a diminishing fraction of the servers during service. The threshold

[2] We use the standard asymptotic notation: $f(n) = o(g(n))$ if $\lim_{n\to\infty} f(n)/g(n) = 0$; $f(n) = \Theta(g(n))$ if $\lim_{n\to\infty} f(n)/g(n)$ is a positive constant; $f(n) = \Omega(g(n))$ if $\lim_{n\to\infty} f(n)/g(n)$ is no smaller than a positive constant.

$\gamma < 1$ is tight in the sense that for a single job-class system, $P_Q^{(n)}$ is not diminishing when $\gamma = 1$. This is easy to see by noting that when $\gamma = 1$, the single job-class system is equivalent to a classical M/M/s system where $s = \lfloor n/m_{\max}^{(n)} \rfloor$ is a constant and the load is also a constant.

- **Region 3:** When the load $\rho^{(n)}$ is in *heavy-traffic* ($\alpha > 0$) and the server needs stay *constant* ($\gamma = 0$), does the queueing probability, $P_Q^{(n)}$, diminish as $n \to \infty$?

  Our result: We show that $P_Q^{(n)}$ is diminishing when the load satisfies $0 < \alpha < 1/2$, which is a traffic regime analogous to the sub-Halfin-Whitt regime in the literature. The threshold $\alpha < 1/2$ is tight in the sense that for a single job-class system, $P_Q^{(n)}$ is not diminishing when $\alpha \geq 1/2$. Again, this can been seen by noting that when $\alpha \geq 1/2$, the single job-class system is equivalent to a classical M/M/s system with $s = \lfloor n/m_{\max}^{(n)} \rfloor = \Theta(n)$, whose traffic is at least as heavy as the Halfin-Whitt regime.

- **Region 4:** When the load $\rho^{(n)}$ is in *heavy-traffic* ($\alpha > 0$) and the server needs also *scale* ($\gamma > 0$), does the queueing probability, $P_Q^{(n)}$, diminish as $n \to \infty$?

  Our result: We show that $P_Q^{(n)}$ is diminishing when $2\alpha + \gamma < 1$. This exact characterization of the joint scaling regime for diminishing queueing probability formalizes the intuition that queueing is negligible in large systems if each job does not need too many servers and the load is not too heavy. This threshold $2\alpha + \gamma < 1$ is also tight in the sense that for a single job-class system, the queueing probability is not diminishing when $2\alpha + \gamma \geq 1$. To see this, we again consider the equivalent M/M/s system with $s = \lfloor n/m_{\max}^{(n)} \rfloor$. It is not hard to verify that the load of this equivalent system is $1 - \Theta(n^{-\alpha}) = 1 - \Theta(s^{-\frac{\alpha}{1-\gamma}})$, which is at least as heavy as Halfin-Whitt since $\frac{\alpha}{1-\gamma} \geq 1/2$ under the condition that $2\alpha + \gamma \geq 1$.

In fact, our characterization of the queueing probability in Theorem 4.2 has the following general form, which not only unifies all of the four regions above but also provides an upper bound on the rate of convergence for the diminishing queueing probability:

$$P_Q^{(n)} \leq \frac{1}{1-\rho^{(n)}} \left(3K \sqrt{\frac{m_{\max}^{(n)}}{n}} + \frac{m_{\max}^{(n)}}{n} \right). \tag{2}$$

Transient analysis for the number of jobs in the system. We also characterize the transient behavior of the system under constant load in the limit as the number of servers n goes to infinity. Specifically, in Theorem 4.3 we show that the sequence of processes for appropriately scaled number of jobs in the system converges to a deterministic system over any finite time interval, as $n \to \infty$. In particular, the deterministic system is the unique solution to a fluid model, which converges to an equilibrium point as $t \to \infty$. Therefore, we have established that, for any sufficiently large *fixed-time* interval, the scaled number of jobs of a large system (large n) is approximated by the equilibrium point of the fluid model. We conjecture that the stationary distribution converges to the equilibrium point.

Technical challenges and our new approach

To better understand the queueing dynamics, it is crucial to characterize how many jobs of each class are in service, since that determines the total departure rate of jobs. However, due to the heterogeneous server needs of jobs, the number of jobs in service turns out to be very hard to analyze. The root of the difficulty is that the process by which jobs enter service can be quite bursty. For example, when a job in service finishes, it could happen that *many* jobs in the queue will be

admitted into service *all at the same time* if the completed job has a high server need; or it could happen that *no job* in the queue will enter service at all if the completed job frees up a small number of servers and the head-of-queue job cannot fit. This "jitter effect" makes it highly challenging to reason about jobs in service.

Our approach is similar in spirit to the drift method (see Section 5). However, our construction of the Lyapunov function cannot be derived using existing methods as in [13, 26–28, 31, 50, 52, 53] and instead requires new insights. In particular, we find a Lyapunov function whose drift has an interesting upper-bounding function, which then allows us to establish an upper bound on the queueing probability. Our Lyapunov function also has an intuitive meaning: it corresponds to the total work contributed by all the jobs in the system. Here the work contributed by a job is the product of its server need and its expected service time. More details on the drift method and the steps in our approach are given in Sections 5.1 and 5.2.

2 RELATED WORKS

Multi-server job models. At present almost nothing is known regarding the performance of the multi-server job model. A few works have sought to derive the steady-state distribution of the number of jobs in the system under a highly simplified model, where all jobs have the same exponential service duration $S_i \sim \text{Exp}(\mu)$, for all classes i, and there are only $n = 2$ servers, see papers by Brill and Green [10] and Filippopoulos and Karatza [14]. However these solutions are highly complex, typically involving roots to a quartic equation; this makes the solutions impractical. Earlier work by Kim [21], based on a matrix analytic approach, is similarly impractical since it scales exponentially with the size of the system. A survey paper by Melikov [34] summarizes these approaches and a few others. In summary, understanding the response time for multi-server job systems with more than $n = 2$ servers is entirely open, even when all jobs have the same exponentially-distributed service durations.

Not only is the response time intractable for the multi-server job model, but even the stability region for this model (under FCFS scheduling) is only partially understood. In 2017, Rumyantsev and Morozov [41] derived the stability region for the multi-server job model where, for all class i, all jobs have the *same* exponential service duration $S_i \sim \text{Exp}(\mu)$. This work was generalized in [36] and [2] to allow for more general arrival processes, still under the assumption that all jobs have the same exponential service duration. Very recently, Grosof et al. [15] derived a simple closed-form expression for the stability region for the multi-server job model where jobs have *different* exponential service time durations. Unfortunately, the work [15] is limited to the case of only two classes. The characterization of stability in the case of more than two job classes is open.

We reiterate that the key technical difficulty is rooted in the heterogeneous server needs of multi-server jobs, rendering classical general frameworks for stability less effective or inapplicable. The sufficient condition (1) that we establish for stability is derived under the framework based on Lyapunov drift (see, e.g., [43], for a comprehensive coverage). Here the heterogeneity in server needs makes it hard to tighten this condition in the non-asymptotic regime. Another general stability framework is based on the saturation rule [4] for systems with certain monotonicity. Roughly speaking, the monotonicity property in this framework requires that when the queueing system has a finite number of job arrivals, delaying job arrival times can only delay the time when all the jobs are completed. The monotonicity property is satisfied by many classical queueing models including the M/M/n model. However, it is not satisfied by the multi-server job model under FCFS. Counterexamples can be constructed where delaying job arrivals actually leads to a shorter overall completion time due to less server idling time.

While very little is known about the performance of multi-server job models, there is a close cousin of the model, called the *dropping model* which is analytically tractable under very general

settings. In the dropping model, those jobs which cannot immediately receive service are simply dropped. When the job durations are exponentially distributed, the stationary distribution of the dropping model exhibits a beautiful product form. Arthurs and Kaufman [3] were the first to observe the product form. Whitt [54] generalized the model to allow jobs to demand multiple resource types (e.g., both CPU and I/O). van Dijk [47] allowed durations to be generally distributed. Tikhonenko [45] combined aspects of [54] and [47].

The multi-server job model is also related to *streaming models* for communication networks. Here the resource being shared is bandwidth in the network. The "jobs" are audio or video flows which require a fixed bandwidth reservation to run (this is akin to needing a fixed number of servers). Flows requiring fixed bandwidth are often referred to as streaming (see [6]), but are also sometimes referred to as "inelastic jobs" (see [33, 38]). The papers dealing with streaming jobs typically operate in the dropping model, where the goal is to schedule to minimize a cost related to dropping probabilities (see [5, 11, 18, 19]). Note that the setting here is more complex than the multi-server job dropping model. The added complexity sometimes comes from the fact that the authors are seeking an optimal dropping policy and sometimes is due to a network setting.

Another class of related models are models for the virtual machine (VM) scheduling problem [30, 32, 39, 40, 56], but again only limited results are available for job response times and most works focus on stability. In the VM scheduling problem, a VM job requests multiple units of resource such as CPUs. But the server model in VM scheduling is different from the multi-server job model we study in this paper. In VM scheduling, a server has several CPUs and usually can accommodate multiple jobs, but a job cannot be spread across multiple servers.

Diminishing queueing probability in other models. The concept of diminishing queueing probability has been studied across a wide class of problems. As already discussed, in the M/M/n model, it was shown that diminishing queueing probability occurs in the sub-Halfin-Whitt regime. Recently, motivated by the desire for low latency in today's computing applications, the interest in diminishing queueing probability (and queueing time) have been greatly renewed. Investigating conditions and policies for achieving diminishing queueing probability has become a rapidly growing research area with a rich body of work.

The queueing probability (and queueing time) under scaling regimes has also been studied in load-balancing models, where each server has its own queue, and where each arriving job is immediately dispatched to a server upon arrival. The Join-the-Idle-Queue (JIQ) policy routes every arriving job to an idle queue, if one exists, otherwise to a randomly selected queue [29, 44]. For JIQ it was shown that diminishing queueing probability can be achieved when the load, $\rho^{(n)}$, is lighter than $1 - \frac{1}{n}$ [25, 26, 28]. Another example is the Power-of-d-Choices policy (Pod), where every arriving job samples d queues and goes to the shortest of these [35, 49]. For Pod it was shown that diminishing queueing probability can be achieved when $d = \Omega\left(\frac{\log n}{1-\rho^{(n)}}\right)$ in the sub-Halfin-Whitt regime and when $d = \Omega\left(\frac{\log^2 n}{1-\rho^{(n)}}\right)$ for $\rho^{(n)}$ lighter than $1 - \frac{1}{n}$ [26–28]. Moreover, the Join-the-Shortest-Queue (JSQ) load-balancing policy can be viewed as a special case of Pod, where $d = n$. Thus it too has diminishing queueing probability. Finally, Weng et al. [53] extend the traditional load-balancing model to address data locality and again prove diminishing queueing probability under specific conditions on the data locality.

Another model where diminishing queueing time has been investigated is the multi-task job model. Here, again, each server has its own queue. However, each job consists of k tasks that can run on servers in parallel with i.i.d. service time requirements, but the job is not completed until all of its tasks complete [52]. The multi-task model is closest to our own, but differs in several ways. Firstly, there are queues at each server, where the tasks of a job need to be dispatched to the queues

Service time	$S_i \sim \text{Exp}(\mu_i)$
Server need	$m_i^{(n)}$
Arrival rate	$\lambda_i^{(n)} := \lambda^{(n)} \cdot p_i^{(n)}$
Class i load	$\rho_i^{(n)} := \frac{\lambda_i^{(n)} m_i^{(n)}}{n\mu_i}$

Table 1. Variables related to class i jobs

Number of servers	n
Number of class i jobs in system	$X_i^{(n)}$
Number of class i jobs in queue	$Q_i^{(n)}$
Classes of jobs in system	$\boldsymbol{U}^{(n)}$

Table 2. Variables related to system states

upon arrival. This implies, importantly, that the tasks of a job typically do not end up executing simultaneously. Secondly, the multi-task model is different from our own in that the individual tasks of a job have i.i.d. service time requirements which may be quite different from each other. Within the multi-task model, Weng and Wang [52] analyze a Batch-Filling-d policy, where each arriving job samples kd queues and fills its tasks into queues so as to minimize the individual task queueing times. For this Batch-Filling-d algorithm, Weng and Wang [52] show that diminishing queueing time for jobs can be achieved in the sub-Halfin-Whitt regime when d is sufficiently high.

3 MODEL

The basic description of the multi-server job queueing model with n servers follows Figure 2. The notation that we have discussed so far appears in Table 1. We define $m_{\max}^{(n)} := \max_{1\le i\le K} m_i^{(n)}$ as the maximum server need, and $\rho^{(n)} := \sum_{i=1}^K \rho_i^{(n)} = \sum_{i=1}^K \frac{\lambda_i^{(n)} m_i^{(n)}}{n\mu_i}$ as the system load. Let $\mu_{\max} := \max_{1\le i\le K} \mu_i$.

Arriving jobs enter a First-Come-First-Served (FCFS) queue, which has an infinite capacity. As illustrated in Figure 2, when a job arrives, it enters service if the queue is empty and the number of idle servers is at least the job's server need; otherwise, the job enters the queue to wait for service.

The state of the multi-server job queueing system can be described by an ordered list of the classes of all jobs in the system, in arrival order. We denote the state at time t by $\boldsymbol{U}^{(n)}(t) = \big(u_1(t), u_2(t), \ldots, u_J(t)\big)$, where $u_j(t)$ is the class of the jth oldest job in the system at time t. For the state shown in Figure 2, assuming that the three jobs in service arrived in the order of 3, then 1, then 4, the state descriptor is given by $(3, 1, 4, 3, 4, 1, 4, 1)$. Note that this state descriptor is *ordered* and the order determines which jobs are in service. Therefore, the total job departure rate depends on the order, which implies that the queue is not an order-independent queue [8, 22]. It can be verified that the continuous-time process, $\boldsymbol{U}^{(n)} = \big\{\boldsymbol{U}^{(n)}(t), t \ge 0\big\}$, forms an irreducible Markov chain taking values in $\mathcal{U} = \{(u_1, u_2, \ldots, u_J) \colon J \in \mathbb{N}, u_j \in \{1, 2, \ldots, K\} \forall j\}$.

Let $X_i^{(n)}(t)$ denote the number of class i jobs in the system at time t for $i = 1, \ldots, K$. Let $\boldsymbol{X}^{(n)}(t)$ be the corresponding vector. Note that the system size process $\boldsymbol{X}^{(n)} = \big\{\boldsymbol{X}^{(n)}(t), t \ge 0\big\}$ is not a Markov chain. We define the queue size process $\boldsymbol{Q}^{(n)} = \big\{\boldsymbol{Q}^{(n)}(t), t \ge 0\big\}$, where $\boldsymbol{Q}^{(n)}(t) = \big(Q_1^{(n)}(t), \ldots, Q_K^{(n)}(t)\big)$ and $Q_i^{(n)}(t)$ is the number of class i jobs waiting in the queue at time t. We summarize the variables related to system state in Table 2.

Observe that each of $\boldsymbol{X}^{(n)}$ and $\boldsymbol{Q}^{(n)}$ is a function of the Markov chain $\boldsymbol{U}^{(n)}$. When the Markov chain $\boldsymbol{U}^{(n)}$ is positive recurrent with a unique stationary distribution, $\boldsymbol{X}^{(n)}$ also has a unique stationary distribution, as does $\boldsymbol{Q}^{(n)}$. We use $\boldsymbol{X}^{(n)}(\infty)$ and $\boldsymbol{Q}^{(n)}(\infty)$ to denote the random variables that have the stationary distribution of $\boldsymbol{X}^{(n)}$ and $\boldsymbol{Q}^{(n)}$, respectively.

In this paper, we are interested in characterizing the queueing probability, under FCFS scheduling, in different scaling regimes. Here we use the term *queueing probability*, denoted by $P_Q^{(n)}$, to refer to the steady-state probability that a job cannot enter service immediately upon arrival. Note that when there are more than or equal to $m_{\max}^{(n)}$ idle servers, an arriving job will enter service

immediately. Thus, we can upper bound $P_Q^{(n)}$ as follows

$$P_Q^{(n)} \leq \mathbb{P}\left(\sum_{i=1}^{K} m_i^{(n)} X_i^{(n)}(\infty) > n - m_{\max}^{(n)}\right). \tag{3}$$

4 MAIN RESULTS

In this section we formally present our main theorems.

Theorem 4.1 (**Stability Condition**). *Consider the system with n servers and K classes of multi-server jobs. Under the FCFS policy, the Markov chain $\boldsymbol{U}^{(n)}$ is positive recurrent, i.e., the system is stable, when the load $\rho^{(n)}$ satisfies*

$$\rho^{(n)} < 1 - \frac{m_{\max}^{(n)}}{n}. \tag{4}$$

Theorem 4.2 (**Diminishing Queueing Probability**). *Consider the system with n servers and K classes of multi-server jobs. Assume that it uses the FCFS policy and the load satisfies the stability condition $\rho^{(n)} < 1 - \frac{m_{\max}^{(n)}}{n}$. Then the queueing probability is upper bounded by:*

$$P_Q^{(n)} \leq \frac{1}{1-\rho^{(n)}}\left(3K\sqrt{\frac{m_{\max}^{(n)}}{n}} + \frac{m_{\max}^{(n)}}{n}\right). \tag{5}$$

Consequently, if the joint scaling of $\rho^{(n)}$ and $m_{\max}^{(n)}$ satisfies $\frac{1}{1-\rho^{(n)}}\sqrt{\frac{m_{\max}^{(n)}}{n}} = o(1)$, then the queueing probability is diminishing, i.e.,

$$\lim_{n\to\infty} P_Q^{(n)} = 0. \tag{6}$$

Finally, we show that the evolution of the scaled number of jobs of each class in the system, $\frac{1}{\lambda_i^{(n)}} X_i^{(n)}(t)$, can be approximated by a deterministic system, over any finite time horizon $[0, T]$. In particular, the deterministic system is governed by the following differential equations:

$$\dot{y}_i(t) = 1 - \min\left\{\mu_i y_i(t), \frac{1}{\rho_i}\right\}, \qquad i \in \{1, \ldots, K\}, \tag{7}$$

Note that when each job can enter service immediately upon arrival, the original system has the same dynamics as an "enlarged" system where each class of jobs has access to a separate set of n servers and is served in FCFS order, i.e., class i jobs run as an M/M/$\frac{n}{m_i^{(n)}}$ queue with arrival rate $\lambda_i^{(n)}$ and service rate μ_i. The diminishing queueing probability implied by Theorem 4.2 suggests that the original system can be approximated by such an enlarged system. On the other hand, we can view the solution $y_i(t)$ to equation (7) as a deterministic approximation to the sample path of the scaled number of class i jobs in the enlarged system. The deterministic system $\boldsymbol{y}(t)$ hence also provides an approximation for the scaled number of jobs in the original system.

Theorem 4.3 (**Transient behavior of the number of jobs in the system**). *Suppose that for each class $i \in \{1, \ldots, K\}$, the load satisfies $\rho_i^{(n)} = \rho_i > 0$ for all n, and $\rho = \sum_{i=1}^{K} \rho_i < 1$. Assume that $\lim_{n\to\infty} \frac{1}{\lambda_i^{(n)}} X_i^{(n)}(0) = y_i(0)$ in probability for each i, where $\boldsymbol{y}(0)$ is a deterministic initial condition such that $0 \leq y_i(0) < \frac{1}{\rho\mu_i}$ for all i. Let $\boldsymbol{y}(t)$ be the unique solution to the differential equation (7) given*

initial condition $\boldsymbol{y}(0)$. *If* $m_{\max}^{(n)}$ *satisfies* $m_{\max}^{(n)} = o(n)$, *then for any fixed* $T > 0$, *the following holds:*

$$\lim_{n\to\infty} \sup_{0\le t\le T} \sum_{i=1}^{K} \left| \frac{1}{\lambda_i^{(n)}} X_i^{(n)}(t) - y_i(t) \right| = 0, \quad \text{in probability.} \tag{8}$$

5 PROOFS FOR STABILITY AND DIMINISHING QUEUEING PROBABILITY

In this section we present the proofs of stability (Theorem 4.1) and diminishing queueing probability (Theorem 4.2). Since the structure of our proofs is similar in spirit to that of the recently-developed drift method [13, 31, 50], we first provide some background on the drift method in Section 5.1. We then sketch our approach in Section 5.2 and highlight how it deviates from the traditional drift method. Finally, we present the detailed proofs in Section 5.3.

5.1 Preliminaries on drift method

The general idea of the drift method is to construct an appropriate Lyapunov function and then study the drift of the Lyapunov function. To be more concrete, consider the Markov chain $\boldsymbol{U}^{(n)}$ that describes the state of our multi-server job queueing system, and let $g\colon \mathcal{U} \to \mathbb{R}_+$ be a Lyapunov (nonnegative) function on the state space. Let $\boldsymbol{u}$ denote a state and $r_{\boldsymbol{u},\boldsymbol{u}'}$ denote the transition rate from a state $\boldsymbol{u}$ to another state $\boldsymbol{u}'$. Then the drift of g is defined as

$$\begin{aligned}\Delta g(\boldsymbol{u}) &= \lim_{\delta\to 0} \frac{1}{\delta} \mathbb{E}\left[g\left(\boldsymbol{U}^{(n)}(t+\delta)\right) - g\left(\boldsymbol{U}^{(n)}(t)\right) \,\middle|\, g\left(\boldsymbol{U}^{(n)}(t)\right) = \boldsymbol{u} \right] \\ &= \sum_{\boldsymbol{u}'\in\mathcal{U}} r_{\boldsymbol{u},\boldsymbol{u}'} \left(g(\boldsymbol{u}') - g(\boldsymbol{u}) \right).\end{aligned}$$

From the definition, it can be seen that the drift Δg is a function of the state $\boldsymbol{u}$. When Δg is applied to the steady state $\boldsymbol{U}^{(n)}(\infty)$, the drift $\Delta g\left(\boldsymbol{U}^{(n)}(\infty)\right)$ is referred to as the *steady-state drift*.

The drift method utilizes the relationship that the expected steady-state drift is zero for well-behaved Lyapunov functions, i.e., $\mathbb{E}\left[\Delta g\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] = 0$. One key to the drift method is to craft an appropriate Lyapunov function g such that its drift, $\Delta g\left(\boldsymbol{U}^{(n)}(\infty)\right)$, decomposes into terms that correspond to the metric of interest (e.g., the total number of jobs in the system) and terms that are tractable to bound. There are mainly two general approaches to constructing such Lyapunov functions. The traditional approach [13, 31, 50] is based on establishing state-space collapse, a property whereby the state is concentrated around a strict subset of the entire state space in heavy load. Another approach [26–28, 52, 53] is based on solving the so-called Stein's equation after coupling the system with a simple fluid model.

However, it is hard to apply either of these two approaches to our problem. The first approach is most effective when the load scaling is in the traditional heavy-traffic regime, where the number of servers stays constant and the load approaches 1; this is not our setting. The second approach requires finding a reasonable fluid model that admits a solution to Stein's equation. Such a fluid representation is hard to find for our "jittery" model, where jobs sometimes enter service in batches.

5.2 Our approach

Our construction of the Lyapunov function g does not fall under either of the two typical approaches in the literature, although our approach follows the general framework of the drift method in the sense that we also exploit the identity $\mathbb{E}\left[\Delta g\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] = 0$.

We consider a Lyapunov function $g\colon \mathcal{U} \to \mathbb{R}_+$ defined as follows:

$$g(\boldsymbol{u}) = \sum_{i=1}^{K} \frac{m_i^{(n)} x_i}{\mu_i}, \tag{9}$$

where x_i denotes the number of class i jobs in system (in queue plus in service), with the understanding that x_i is a function of the state $\boldsymbol{u}$. Note that $g(\boldsymbol{u})$ has the following intuitive meaning: Since each class i job needs to occupy $m_i^{(n)}$ servers for an average of $1/\mu_i$ duration of time, we say that the *expected work* contributed by a class i job, measured by this space-time product, is $m_i^{(n)}/\mu_i$. Then $g(\boldsymbol{u})$ represents the total amount of expected work in the system.

We then extract information from the drift $\Delta g(\boldsymbol{u})$ in the following way. We first derive the following upper-bounding function on $\Delta g(\boldsymbol{u})$, which will be formally stated as Lemma 5.1 in Section 5.3:

$$\Delta g(\boldsymbol{u}) \le m_{\max}^{(n)} + \begin{cases} n\rho^{(n)} - \sum_{i=1}^K m_i^{(n)} x_i, & \text{if } \sum_{i=1}^K m_i^{(n)} x_i \le n - m_{\max}^{(n)}, \\ -n\left(1-\rho^{(n)}\right), & \text{if } \sum_{i=1}^K m_i^{(n)} x_i > n - m_{\max}^{(n)}. \end{cases}$$

Observe that the second term in the upper-bounding function corresponds to the condition $\sum_{i=1}^K m_i^{(n)} x_i > n - m_{\max}^{(n)}$. This will allow us to relate $\mathbb{P}\left(\sum_{i=1}^K m_i^{(n)} X_i^{(n)}(\infty) > n - m_{\max}^{(n)}\right)$ to $\mathbb{E}\left[\Delta g\left(\boldsymbol{U}^{(n)}(\infty)\right)\right]$, where recall that $X_i^{(n)}(\infty)$ denotes the number of class i jobs in steady state. Then utilizing the identity $\mathbb{E}\left[\Delta g\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] = 0$ will eventually lead to an upper bound on the queueing probability $P_Q^{(n)}$.

5.3 Proofs

Consider the Lyapunov function g defined in (9) and let $\Delta g(\boldsymbol{u})$ denote its drift. We organize our proofs into three parts: we first establish the upper-bounding function on $\Delta g(\boldsymbol{u})$ in Lemma 5.1, which underpins all of our analysis; we then prove the stability result (Theorem 4.1) based on Lemma 5.1; finally we prove the results on queueing probability (Theorem 4.2) through several more lemmas. The flow chart of the proofs is given in Figure 4.

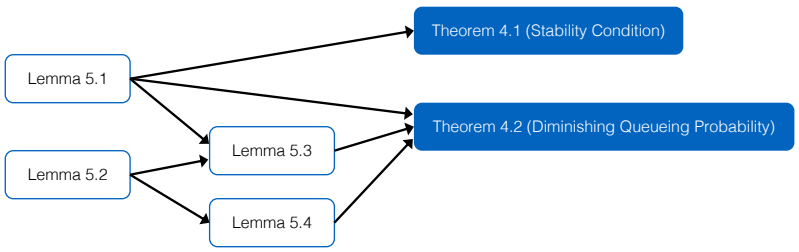

Fig. 4. Flowchart of proofs

Upper-bounding function on the drift.

Lemma 5.1. *The drift of $g(\boldsymbol{u})$ can be bounded as:*

$$\Delta g(\boldsymbol{u}) \le m_{\max}^{(n)} + h(\boldsymbol{x}), \tag{10}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$ with x_i denoting the number of class i jobs in system (in queue plus in service), and the function h is defined as:

$$h(\boldsymbol{x}) = \begin{cases} n\rho^{(n)} - \sum_{i=1}^K m_i^{(n)} x_i, & \text{if } \sum_{i=1}^K m_i^{(n)} x_i \le n - m_{\max}^{(n)}, \\ -n\left(1-\rho^{(n)}\right), & \text{if } \sum_{i=1}^K m_i^{(n)} x_i > n - m_{\max}^{(n)}. \end{cases} \tag{11}$$

Proof. Recall that $g(\boldsymbol{u}) = \sum_{i=1}^{K} \frac{m_i^{(n)} x_i}{\mu_i}$. Let q_i be the corresponding number of class i jobs in the queue, with the understanding that q_i is a function of the state $\boldsymbol{u}$. When there is a class i arrival, which happens at a transition rate of $\lambda_i^{(n)}$, the value of x_i increases by 1; when a class i job departs, which happens at a transition rate of $\mu_i(x_i - q_i)$, the value of x_i decreases by 1. Therefore, the drift of $g(\boldsymbol{u})$ can be written as follows:

$$\Delta g(\boldsymbol{u}) = \sum_{i=1}^{K} \lambda_i^{(n)} \cdot \frac{m_i^{(n)}}{\mu_i} - \sum_{i=1}^{K} \mu_i (x_i - q_i) \cdot \left(-\frac{m_i^{(n)}}{\mu_i} \right).$$

Then noticing the relation $\rho_i^{(n)} = \frac{\lambda_i^{(n)} m_i^{(n)}}{n\mu_i}$, we get

$$\Delta g(\boldsymbol{u}) = n\rho^{(n)} - \sum_{i=1}^{K} m_i^{(n)} (x_i - q_i). \tag{12}$$

Consider the term $\sum_{i=1}^{K} m_i^{(n)} (x_i - q_i)$ in (12). It is easy to see that when $\sum_{i=1}^{K} m_i^{(n)} x_i \le n - m_{\max}^{(n)}$, which corresponds to the *first case* in the definition of h in (11), there must be no jobs in the queue, and thus $\sum_{i=1}^{K} m_i^{(n)} (x_i - q_i) = \sum_{i=1}^{K} m_i^{(n)} x_i$. When $\sum_{i=1}^{K} m_i^{(n)} x_i > n - m_{\max}^{(n)}$, which corresponds to the *second case* in the definition of h in (11), either there are no jobs in the queue and thus $\sum_{i=1}^{K} m_i^{(n)} (x_i - q_i) = \sum_{i=1}^{K} m_i^{(n)} x_i > n - m_{\max}^{(n)}$, or the number of idle servers is not enough to absorb the job at the head of the queue and thus $\sum_{i=1}^{K} m_i^{(n)} (x_i - q_i) > n - m_{\max}^{(n)}$. In both scenarios, $\Delta g(\boldsymbol{u}) \le n\rho^{(n)} - n + m_{\max}^{(n)} = h(\boldsymbol{x}) + m_{\max}^{(n)}$, which completes the proof of Lemma 5.1. □

Proof of Theorem 4.1 (Stability Condition). We invoke the Foster-Lyapunov criteria [43] to show that the Markov chain $\boldsymbol{U}^{(n)}$ is positive recurrent. Let $\mathcal{B} = \left\{ \boldsymbol{u} \colon \sum_{i=1}^{K} m_i^{(n)} x_i \le n - m_{\max}^{(n)} \right\}$. Then clearly $\mathcal{B} \subseteq \mathcal{U}$ is a finite set. The drift upper bound in Lemma 5.1 implies that:

- $\Delta g(\boldsymbol{u}) \le n\rho^{(n)} + m_{\max}^{(n)} < +\infty$ when $\boldsymbol{u} \in \mathcal{B}$;
- $\Delta g(\boldsymbol{u}) \le -n\left(1 - \rho^{(n)}\right) + m_{\max}^{(n)} < 0$ when $\boldsymbol{u} \notin \mathcal{B}$,

where the second item follows from the assumption that $\rho^{(n)} < 1 - \frac{m_{\max}^{(n)}}{n}$. Therefore, by the Foster-Lyapunov theorem, the Markov chain $\boldsymbol{U}^{(n)}$ is positive recurrent. □

Proof of Theorem 4.2 (Diminishing Queueing Probability). The proof of Theorem 4.2 is based on two more lemmas (Lemmas 5.3 and 5.4) besides Lemma 5.1. To keep it clear and short, we defer the proofs of Lemmas 5.3 and 5.4 until the end.

First we note that $\mathbb{E}\left[\Delta g\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] = 0$ since $\mathbb{E}\left[g\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] < +\infty$ by Lemma 5.3. Then utilizing $\mathbb{E}\left[\Delta g\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] = 0$, we can derive an upper bound on $\mathbb{P}\left(\sum_{i=1}^{K} m_i^{(n)} X_i^{(n)}(\infty) > n - m_{\max}^{(n)}\right)$, where recall that the queueing probability $P_Q^{(n)} \le \mathbb{P}\left(\sum_{i=1}^{K} m_i^{(n)} X_i^{(n)}(\infty) > n - m_{\max}^{(n)}\right)$. By the drift upper bound $\Delta g(\boldsymbol{u}) \le h(\boldsymbol{x}) + m_{\max}^{(n)}$ in Lemma 5.1, we have

$$\mathbb{E}\left[h\left(\boldsymbol{X}^{(n)}(\infty)\right)\right] \ge \mathbb{E}\left[\Delta g\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] - m_{\max}^{(n)} = -m_{\max}^{(n)}.$$

Since $\left[h\left(\boldsymbol{X}^{(n)}(\infty)\right)\right]$ can be written as follows, by its construction:

$$\mathbb{E}\left[h\left(\boldsymbol{X}^{(n)}(\infty)\right)\right] = \mathbb{E}\left[\left(n\rho^{(n)} - \sum_{i=1}^{K} m_i^{(n)} X_i^{(n)}(\infty)\right) \cdot \mathbb{1}_{\left\{\sum_{i=1}^{K} m_i^{(n)} X_i^{(n)}(\infty) \le n - m_{\max}^{(n)}\right\}}\right]$$

$$- n\left(1-\rho^{(n)}\right)\mathbb{P}\left(\sum_{i=1}^{K} m_i^{(n)} X_i^{(n)}(\infty) > n - m_{\max}^{(n)}\right),$$

it follows that

$$\begin{aligned}
&\mathbb{P}\left(\sum_{i=1}^{K} m_i^{(n)} X_i^{(n)}(\infty) > n - m_{\max}^{(n)}\right) \\
&\le \frac{m_{\max}^{(n)}}{n\left(1-\rho^{(n)}\right)} + \frac{1}{n\left(1-\rho^{(n)}\right)}\mathbb{E}\left[\left(n\rho^{(n)} - \sum_{i=1}^{K} m_i^{(n)} X_i^{(n)}(\infty)\right)\cdot \mathbb{1}_{\left\{\sum_{i=1}^{K} m_i^{(n)} X_i^{(n)}(\infty) \le n - m_{\max}^{(n)}\right\}}\right]. \\
&\le \frac{m_{\max}^{(n)}}{n\left(1-\rho^{(n)}\right)} + \frac{1}{n\left(1-\rho^{(n)}\right)}\sum_{i=1}^{K}\mathbb{E}\left[\left(n\rho_i^{(n)} - m_i^{(n)} X_i^{(n)}(\infty)\right)^+\right], \qquad (13)
\end{aligned}$$

where $(a)^+$ for a real number $a \in \mathbb{R}$ denotes $\max\{a, 0\}$.

We then bound each summand in the second term in (13) using Lemma 5.4, which asserts that $\mathbb{E}\left[\left(n\rho_i^{(n)} - m_i^{(n)} X_i^{(n)}(\infty)\right)^+\right] \le 3\sqrt{n m_i^{(n)}}$. Intuitively, Lemma 5.4 says that the number of class i jobs, $X_i^{(n)}(\infty)$, cannot be much smaller than $\frac{n\rho_i^{(n)}}{m_i^{(n)}} = \frac{\lambda_i^{(n)}}{\mu_i}$. To see this, consider the scenario when the number of class i jobs is smaller than $\frac{\lambda_i^{(n)}}{\mu_i}$. Then the departure rate of class i jobs will definitely be much smaller than $\mu_i \cdot \frac{\lambda_i^{(n)}}{\mu_i} = \lambda_i^{(n)}$, i.e., the departure rate is smaller than the arrival rate for class i jobs. Consequently, the number of class i jobs will increase. The threshold value for the number of class i jobs to balance the departure and arrival rates is $\frac{\lambda_i^{(n)}}{\mu_i}$. Conceptually, Lemma 5.4 is similar to the "state-space collapse" type of results in the literature, since it can be interpreted as a concentration of $X_i^{(n)}(\infty)$ around the subset of values no smaller than $\frac{\lambda_i^{(n)}}{\mu_i}$.

With this upper bound given by Lemma 5.4, (13) can be further upper bounded as:

$$\begin{aligned}
\mathbb{P}\left(\sum_{i=1}^{K} m_i^{(n)} X_i^{(n)}(\infty) > n - m_{\max}^{(n)}\right) &\le \frac{m_{\max}^{(n)}}{n\left(1-\rho^{(n)}\right)} + \frac{1}{n\left(1-\rho^{(n)}\right)}\cdot 3K\sqrt{n m_{\max}^{(n)}} \\
&= \frac{1}{1-\rho^{(n)}}\left(3K\sqrt{\frac{m_{\max}^{(n)}}{n}} + \frac{m_{\max}^{(n)}}{n}\right).
\end{aligned}$$

Then the diminishing queueing probability result in (6) follows immediately. This completes the proof of Theorem 4.2. □

Lemmas 5.3 and 5.4 (needed in the proof of Theorem 4.2). Before we present Lemmas 5.3 and 5.4, we first state Lemma 5.2 below, which is the tool we use in the proofs of Lemmas 5.3 and 5.4 (see the flowchart in Figure 4). Lemma 5.2 is a well-known result that bounds tail probabilities and moments using drift conditions [7, 16, 50]. We include it here for completeness, and the form below slightly generalizes the commonly used form in the literature.

Lemma 5.2 (Bounds via Drift). *Let $\{S(t), t \ge 0\}$ be a continuous-time Markov chain on a countable state space $\mathcal{S}$ and $r_{s,s'}$ be its transition rate from state s to state s'. Assume that it has a unique stationary distribution and let $S(\infty)$ be a random element that follows the stationary distribution. Let $V : \mathcal{S} \to \mathbb{R}_+$ be a Lyapunov function. Suppose*

$$v_{\max} := \sup_{s,s':\, r_{s,s'}>0} |V(s') - V(s)| < +\infty, \qquad \sup_{s}\sum_{s'} r_{s,s'} < +\infty,$$

and let

$$\delta_{\max} = \sup_{s} \sum_{s':\, V(s')>V(s)} r_{s,s'}|V(s') - V(s)|.$$

Suppose that there exist $B > 0$ and $\gamma > 0$ such that for any s with $V(s) > B$,

$$\Delta V(s) \le -\gamma.$$

Then for all nonnegative integers m,

$$\mathbb{P}\Big(V\big(S(\infty)\big) > B + 2mv_{\max}\Big) \le \left(\frac{\delta_{max}}{\delta_{max}+\gamma}\right)^{m+1}.$$

Further,

$$\mathbb{E}\Big[V\big(S(\infty)\big)\Big] \le B + \frac{2v_{\max}\delta_{\max}}{\gamma}.$$

Now we are ready for Lemmas 5.3 and 5.4 and their proofs.

Lemma 5.3. *The Lyapunov function g defined in (9) satisfies*

$$\mathbb{E}\left[g\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] < +\infty.$$

Proof. We prove Lemma 5.3 by applying Lemma 5.2 to the Lyapunov function g. Recall that $g(\boldsymbol{u}) = \sum_{i=1}^{K} \frac{m_i^{(n)} x_i}{\mu_i}$, where we use $\boldsymbol{u}$ to denote a state and $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$ to denote the corresponding numbers of jobs in the system of each class.

Let $B = \frac{n - m_{\max}^{(n)}}{\mu_{\min}}$. We bound the drift $\Delta g(\boldsymbol{u})$ when $g(\boldsymbol{u}) > B$ using Lemma 5.1. When $g(\boldsymbol{u}) > B$, we have

$$\sum_{i=1}^{K} m_i^{(n)} x_i \ge \mu_{\min} \cdot \sum_{i=1}^{K} \frac{m_i^{(n)} x_i}{\mu_i} = \mu_{\min} \cdot g(\boldsymbol{u}) > n - m_{\max}^{(n)}.$$

Then Lemma 5.1 implies that $\Delta g(\boldsymbol{u}) \le -n\left(1-\rho^{(n)}\right) + m_{\max}^{(n)}$, which is negative when $\rho^{(n)} < 1 - \frac{m_{\max}^{(n)}}{n}$.

Note that when we take the Lyapunov function in Lemma 5.2 to be g, then $v_{\max} = \max_i \left\{\frac{m_i^{(n)}}{\mu_i}\right\} \le \frac{m_{\max}^{(n)}}{\mu_{\min}}$, and $\delta_{\max} = \sum_{i=1}^{K} \frac{m_i^{(n)} \lambda_i^{(n)}}{\mu_i} = n\rho^{(n)}$. Applying Lemma 5.2, we get

$$\mathbb{E}\left[g\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] \le B + \frac{2v_{\max}\delta_{\max}}{n\left(1-\rho^{(n)}\right) - m_{\max}^{(n)}} \le \frac{n - m_{\max}^{(n)}}{\mu_{\min}} + 2 \cdot \frac{m_{\max}^{(n)}}{\mu_{\min}} \cdot \frac{n\rho^{(n)}}{n\left(1-\rho^{(n)}\right) - m_{\max}^{(n)}},$$

which is finite, and thus completes the proof of Lemma 5.3. □

Lemma 5.4. *For each class i,*

$$\mathbb{E}\left[\left(n\rho_i^{(n)} - m_i^{(n)} X_i^{(n)}(\infty)\right)^+\right] \le 3\sqrt{n m_i^{(n)}}. \tag{14}$$

Proof. We prove Lemma 5.4 by applying Lemma 5.2 to the Lyapunov function $f\colon \mathcal{U} \to \mathbb{R}_+$ defined as

$$f(\boldsymbol{u}) = \left(n\rho_i^{(n)} - m_i^{(n)} x_i\right)^+,$$

where recall that we use $\boldsymbol{u}$ to denote a state and $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$ to denote the corresponding numbers of jobs in the system of each class. Then (14) is equivalent to $\mathbb{E}\left[f\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] \le 3\sqrt{n m_i^{(n)}}$.

Let $B = \sqrt{nm_i^{(n)}}$. Then we bound the drift $\Delta f(\boldsymbol{u})$ when $f(\boldsymbol{u}) > B$. The condition that $f(\boldsymbol{u}) > B$ guarantees that $f(\boldsymbol{u}) = n\rho_i^{(n)} - m_i^{(n)} x_i > B \geq m_i^{(n)}$. Therefore, the drift $\Delta f(\boldsymbol{u})$ can be bounded as follows:

$$\begin{aligned}\Delta f(\boldsymbol{u}) &= \lambda_i^{(n)} \cdot \left(-m_i^{(n)}\right) + \mu_i(x_i - q_i) \cdot m_i^{(n)} \\ &\leq -\mu_i \left(n\rho_i^{(n)} - m_i^{(n)} x_i\right) \\ &< -\mu_i B,\end{aligned}$$

where the last line follows from the condition that $f(\boldsymbol{u}) > B$. Thus, the γ in Lemma 5.2 equals $\mu_i B$.

Note that when we take the Lyapunov function in Lemma 5.2 to be f, then $v_{\max} = m_i^{(n)}$, and $\delta_{\max}$ satisfies $\delta_{\max} \leq \frac{n\mu_i}{m_i^{(n)}} \cdot m_i^{(n)} = n\mu_i$, since f increases when there is a departure of class i jobs. Applying Lemma 5.2, we get

$$\mathbb{E}\left[f\left(\boldsymbol{U}^{(n)}(\infty)\right)\right] \leq B + \frac{2nm_i^{(n)}\mu_i}{\mu_i B} = 3\sqrt{nm_i^{(n)}},$$

which proves the bound (14) in Lemma 5.4. □

6 PROOF FOR TRANSIENT ANALYSIS

In this section, we focus on analyzing the transient behavior of the number of jobs of each class (Theorem 4.3). Our analysis is motivated by the approach of *fluid approximation*, where an appropriately scaled system process can be approximated, as the number of servers grows large, by a deterministic system that is defined by a system of differential equations. Such a deterministic system is referred to as *fluid model* in the literature. In particular, our proof is closely related to the argument for a result called Kurtz's theorem on density-dependent Markov processes (see Chapter 8 of [23] or Chapter 5.3 of [12]). However, we remark that there are two key differences between our proof and the standard argument. First, we scale the number of jobs of each class by the corresponding arrival rate; in contrast, traditional fluid approximation considers scaling by the number of servers. The traditional fluid scaling does not work for a system with multi-server jobs, where server needs can potentially grow with the number of servers n. Second, because we have multiple job classes, instead of directly comparing our original system with the fluid model, we need to construct an intermediate system and couple it with our original system. We establish the fluid approximation by showing that the intermediate system is close to both the original system and the fluid model.

We first introduce the fluid model and its properties in Section 6.1, and then prove Theorem 4.3 in Section 6.2. For ease of exposition, we denote the scaled number of class i jobs in the system at time t by

$$\widetilde{X}_i^{(n)}(t) := \frac{X_i^{(n)}(t)}{\lambda_i^{(n)}}.$$

6.1 Fluid model

Recall that the fluid model is defined by the following differential equations:

$$\dot{y}_i(t) = 1 - \min\left\{\mu_i y_i(t), \frac{1}{\rho_i}\right\}, \qquad i \in \{1, \ldots, K\}. \tag{15}$$

For each $i \in \{1, \ldots, K\}$, given an initial condition satisfying $y_i(0) \le \frac{1}{\mu_i \rho_i}$, it is not hard to see that the unique solution to the differential equation (15) is given by

$$y_i(t) = \left(y_i(0) - \frac{1}{\mu_i}\right) e^{-\mu_i t} + \frac{1}{\mu_i}. \tag{16}$$

Observe that starting from any initial condition $\boldsymbol{y}(0)$ satisfying $0 \le y_i(0) \le \frac{1}{\mu_i \rho_i}$ for all i, as $t \to \infty$, the system $\boldsymbol{y}(t)$ converges to the following equilibrium point

$$\boldsymbol{y}(\infty) := \left(\frac{1}{\mu_1}, \frac{1}{\mu_2}, \ldots, \frac{1}{\mu_K}\right).$$

Interpretation of the differential equations. Consider a system where each class of jobs has access to a separate set of n servers and is served in FCFS order, i.e., class i jobs run as an M/M/$\frac{n}{m_i^{(n)}}$ queue with arrival rate $\lambda_i^{(n)}$ and service rate μ_i. For simplicity, assume that $n/m_i^{(n)}$ is an integer. Let $Y_i^{(n)}(t)$ denote the number of jobs in this system at time t.

We can view a solution to the fluid model (15), $y_i(t)$, as a deterministic approximation to the sample paths of $\frac{Y_i^{(n)}(t)}{\lambda_i^{(n)}}$ for a large n. Note that $\frac{Y_i^{(n)}(t)}{\lambda_i^{(n)}}$ decreases by $\frac{1}{\lambda_i^{(n)}}$ at rate $\min\left\{\mu_i Y_i^{(n)}(t), \mu_i \frac{n}{m_i^{(n)}}\right\}$, due to the completion of a job. Multiplying the departure rate by the decrease due to a departure, and taking the limit $n \to \infty$, we obtain the second drift term $\min\left\{\mu_i y_i(t), \frac{1}{\rho_i}\right\}$ in (15). The first drift term, 1, corresponds to arrivals, as $\frac{Y_i^{(n)}(t)}{\lambda_i^{(n)}}$ increases by $\frac{1}{\lambda_i^{(n)}}$ at rate $\lambda_i^{(n)}$.

6.2 Proof outline of Theorem 4.3

Here we provide a proof outline of Theorem 4.3 and highlight the difference from the standard argument for Kurtz's theorem. We defer the detailed proof to Appendix A.

As we mentioned earlier, a key difference between our proof and the standard argument is the construction of an intermediate system. In particular, we consider the system $\{\boldsymbol{Y}^{(n)}(t)\}$ constructed in Section 6.1, i.e., $Y_i^{(n)}(t)$ is the number of jobs in an M/M/$\frac{n}{m_i^{(n)}}$ queue at time t. We couple this enlarged system with our system so that they have the same initial state, identical job arrival sequence and identical job service times. With this coupling, the system $\{\boldsymbol{Y}^{(n)}(t)\}$ is identical to our system (in terms of the number of present jobs of each class) until the moment when the total number of servers requested by present jobs exceeds n. Specifically, for any positive time T, we have $X_i^{(n)}(t) = Y_i^{(n)}(t)$ for all classes i and all t with $0 \le t \le T$ if

$$\sup_{0 \le t \le T} \sum_{i=1}^{K} m_i^{(n)} Y_i^{(n)}(t) \le n.$$

To establish that the scaled number of jobs in our original system, $\left\{\frac{1}{\lambda_i^{(n)}} X_i^{(n)}(t)\right\}$, can be approximated by the fluid model $\boldsymbol{y}(t)$, it suffices to show that the scaled number of jobs in the intermediate enlarged system, $\left\{\frac{1}{\lambda_i^{(n)}} Y_i^{(n)}(t)\right\}$, is close to both of $\left\{\frac{1}{\lambda_i^{(n)}} X_i^{(n)}(t)\right\}$ and the fluid model $\boldsymbol{y}(t)$. Specifically, note that

$$\mathbb{P}\left(\sup_{0 \le t \le T} \sum_{i=1}^{K} \left|\frac{1}{\lambda_i^{(n)}} X_i^{(n)}(t) - y_i(t)\right| > \epsilon\right) \le \sum_{i=1}^{K} \mathbb{P}\left(\sup_{0 \le t \le T} \left|\frac{1}{\lambda_i^{(n)}} Y_i^{(n)}(t) - y_i(t)\right| > \frac{\epsilon}{K}\right) \tag{17}$$

$$+ \mathbb{P}\left(\exists\, i \text{ and } t \in [0, T] \text{ s.t. } X_i^{(n)}(t) \ne Y_i^{(n)}(t)\right). \tag{18}$$

Our proof consists of the following three main steps:

Step 1: We show that $\left\{\frac{1}{\lambda_i^{(n)}} Y_i^{(n)}(t)\right\}$ is close to the fluid model $\boldsymbol{y}(t)$. In particular, we upper bound (17) by showing that for each i,

$$\begin{aligned}&\mathbb{P}\left(\sup_{0\le t\le T}\left|\frac{1}{\lambda_i^{(n)}} Y_i^{(n)}(t) - y_i(t)\right| > \frac{\epsilon}{K}\right)\\ \le&\mathbb{P}\left(\left|\frac{1}{\lambda_i^{(n)}} Y_i^{(n)}(0) - y_i(0)\right| > \frac{\epsilon e^{-\mu_i T}}{3K}\right) + 2e^{-\lambda_i^{(n)} T\cdot h\left(\frac{\epsilon e^{-\mu_i T}}{3KT}\right)} + 2e^{-\frac{n\mu_i T}{m_i^{(n)}} h\left(\frac{\rho_i \epsilon e^{-\mu_i T}}{3KT}\right)},\end{aligned} \tag{19}$$

using ideas similar to those used in proving Kurtz's theorem [12, 23].

Step 2: We next show that $Y_i^{(n)}(t)$ is close to $X_i^{(n)}(t)$. Specifically, we upper bound the probability of $X_i^{(n)}(t)$ deviating from $Y_i^{(n)}(t)$ in (18) utilizing the result from Step 1.

Step 3: We combine the results from Step 1-2 to show that

$$\lim_{n\to\infty} \mathbb{P}\left(\sup_{0\le t\le T}\sum_{i=1}^{K}\left|\frac{1}{\lambda_i^{(n)}} X_i^{(n)}(t) - y_i(t)\right| > \epsilon\right) = 0,$$

thus completing the proof of Theorem 4.3.

Remark 1. Theorem 4.3 and the convergence of $\boldsymbol{y}(t)$ to $\boldsymbol{y}(\infty)$ give the following result:

$$\widetilde{X}^{(n)}(t) \xrightarrow{d} \boldsymbol{y}(t), \text{ as } n\to\infty, \qquad \boldsymbol{y}(t)\to\boldsymbol{y}(\infty), \text{ as } t\to\infty, \tag{20}$$

where "$\xrightarrow{d}$" denotes convergence in distribution. On the other hand, when the system load ρ satisfies $\rho < 1 - \frac{m_{\max}^{(n)}}{n}$, Theorem 4.1 implies that $\widetilde{\boldsymbol{X}}^{(n)}$ has a unique stationary distribution. That is, for any n,

$$\widetilde{X}^{(n)}(t) \xrightarrow{d} \widetilde{\boldsymbol{X}}^{(n)}(\infty), \text{ as } t\to\infty,$$

where $\widetilde{\boldsymbol{X}}^{(n)}(\infty)$ is the random variable with the stationary distribution of $\widetilde{\boldsymbol{X}}^{(n)}$. We may expect that equation (20) still holds if we change the order in which the limits over t and n are taken, i.e.,

$$\widetilde{X}^{(n)}(t) \xrightarrow{d} \widetilde{\boldsymbol{X}}^{(n)}(\infty), \text{ as } t\to\infty, \qquad \widetilde{\boldsymbol{X}}^{(n)}(\infty) \xrightarrow{d} \boldsymbol{y}(\infty), \text{ as } n\to\infty. \tag{21}$$

Unfortunately, existing techniques fall short of establishing such an *interchange of limits*. However, we conjecture that $\{\widetilde{\boldsymbol{X}}^{(n)}(\infty)\}_n$ converges to $\boldsymbol{y}(\infty)$ as $n\to\infty$. In particular, our numerical experiments appear to support our conjecture (see Section 7). We leave as an intriguing open question establishing the convergence of $\{\widetilde{\boldsymbol{X}}^{(n)}(\infty)\}_n$.

7 SIMULATION RESULTS

In this section, we perform three sets of simulation experiments to demonstrate our theoretical results and investigate gaps in the theory.

In all the experiments, we simulate a sequence of systems with $n = 2^6, 2^8, 2^{10}, 2^{12}, 2^{14}, 2^{16}$ servers. Since the scaling of jobs' server needs is a distinctive feature of our model, we first focus on a setting with three types of jobs whose server needs are $m_1^{(n)} = 3$, $m_2^{(n)} = \log_2 n$ and $m_3^{(n)} = \sqrt{n}$ for sets I and II; then we vary the maximum server need in set III. The service times are exponentially distributed with rates $\mu_1 = 0.25, \mu_2 = 0.5$ and $\mu_3 = 1$. Values of parameters for sets I and II are summarized in Table 3.

Set I. In the first set of experiments, our goal is to demonstrate the diminishing queueing probability and to investigate the distribution of the number of jobs from each class. Recall that the scaling regimes where we prove a diminishing queueing probability are the four regions in Figure 3. Here we pick the most interesting region, Region 4, where both the server needs and the system load $\rho^{(n)}$ scale with n. Specifically, we choose the load to be $\rho^{(n)} = 1 - \frac{1}{4}n^{-0.1}$. One can verify that this setting satisfies the condition for a diminishing queueing probability established by Theorem 4.2.

Queueing probability. Figure 5 shows the fraction of jobs that have to queue upon arrival for each job class, which serves as an estimate of the queueing probability. To see how reliable these estimates are, we take the last 100, 0000 data points (seen by the Poisson arrivals) and divide them into 10 segments. We then calculate the fraction of jobs that queue for each segment. The points on the curves are the mean fractions averaged over the 10 segments, and the error bars mark one standard deviation. The small error bars in Figure 5 indicate that the estimated queueing probabilities are very stable. The trends of the curves in Figure 5 demonstrate that the queueing probabilities are diminishing as n increases, as predicted by our theoretical results.

Number of jobs in system. Although we were not able to theoretically analyze the number of jobs from each class in steady state, we conjecture that the number converges in distribution based on the transient analysis. Specifically, recall that $X_i^{(n)}(\infty)$ denotes the number of class i jobs (in queue plus in service) in steady state. Then we conjecture that $\frac{1}{\lambda_i}X_i^{(n)}(\infty) \to \frac{1}{\mu_i}$ in distribution. Our simulation results in Figure 6 support this conjecture. In Figure 6, the points on the curves are the average of $\frac{1}{\lambda_i}X_i^{(n)}(t)$ seen by Poisson arrivals when the system is empirically steady, and the error bars mark one empirical standard deviation. Recall that $\mu_1 = 0.25, \mu_2 = 0.5$ and $\mu_3 = 1$. We can see that the curves converge nicely to the point mass distributions at the $\frac{1}{\mu_i}$'s.

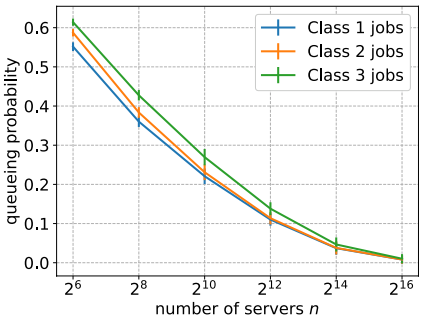

Fig. 5. Diminishing queueing probabilities. System load: $\rho^{(n)} = 1 - \frac{1}{4}n^{-0.1}$; server needs: $\left(m_1^{(n)}, m_2^{(n)}, m_3^{(n)}\right) = \left(3, \log_2 n, \sqrt{n}\right)$.

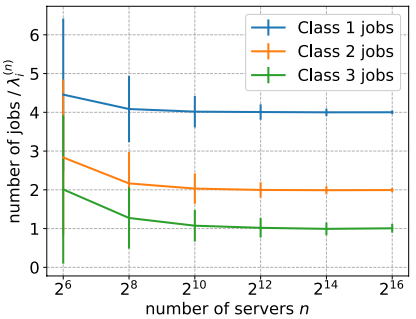

Fig. 6. Convergence of scaled numbers of jobs to $\frac{1}{\mu_i}$, where $\mu_1 = 0.25, \mu_2 = 0.5$, and $\mu_3 = 1$. System load: $\rho^{(n)} = 1 - \frac{1}{4}n^{-0.1}$; server needs: $\left(m_1^{(n)}, m_2^{(n)}, m_3^{(n)}\right) = \left(3, \log_2 n, \sqrt{n}\right)$.

n	$2^6 = 64$	$2^8 = 256$	$2^{10} = 1024$	$2^{12} = 4096$	$2^{14} = 16384$	$2^{16} = 65536$
I, II: $\left(m_1^{(n)}, m_2^{(n)}, m_3^{(n)}\right)$	(3, 6, 8)	(3, 8, 16)	(3, 10, 32)	(3, 12, 64)	(3, 14, 128)	(3, 16, 256)
I: $\rho^{(n)} = 1 - \frac{1}{4}n^{-0.1}$	0.8351	0.8564	0.8750	0.8912	0.9053	0.9175
II: $\rho^{(n)} = 1 - n^{-0.3}$	0.7128	0.8105	0.8750	0.9175	0.9456	0.9641

Table 3. Simulation parameters

Fig. 7. Non-diminishing queueing probabilities. System load: $\rho^{(n)} = 1 - n^{-0.3}$; server needs: $\left(m_1^{(n)}, m_2^{(n)}, m_3^{(n)}\right) = \left(3, \log_2 n, \sqrt{n}\right)$.

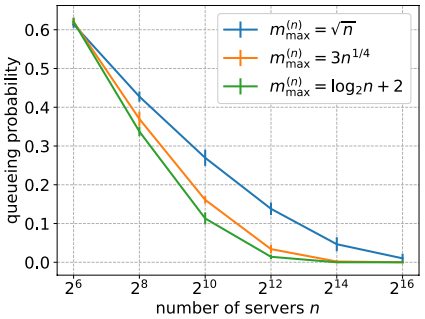

Fig. 8. Impact of maximum server need $m_{\max}^{(n)}$. System load: $\rho^{(n)} = 1 - \frac{1}{4}n^{-0.1}$; server needs: $\left(m_1^{(n)}, m_2^{(n)}, m_3^{(n)}\right) = \left(3, \log_2 n, \sqrt{n}\right)$, $\left(3, \log_2 n, 3n^{1/4}\right)$, $(3, \log_2 n, \log_2 n + 2)$.

Set II. In the second set of experiments, our goal is to empirically investigate *when* the queueing probabilities are not diminishing. We keep the server needs the same as those in the first set of experiments, but we increase the load to be $\rho^{(n)} = 1 - n^{-0.3}$. This scaling regime no longer satisfies our condition for a diminishing queueing probability in Theorem 4.2. Figure 7 demonstrates that indeed, it is unlikely that the queueing probabilities will converge to zero here.

Set III. In the last set of experiments, we investigate how the maximum server need, $m_{\max}^{(n)}$, affects the queueing probability. Figure 8 compares the queueing probabilities of class 3 jobs under three settings where $m_{\max}^{(n)}$ varies as $\sqrt{n}$, $3n^{1/4}$, and $\log_2 n + 2$, while keeping other parameters the same. It shows that the queueing probability diminishes faster as $m_{\max}^{(n)}$ decreases.

8 CONCLUSION AND FUTURE WORK

In this paper, we consider a model that we refer to as the multi-server job queueing model. In this model, a job requests multiple servers and holds on these servers simultaneously during its service. We investigate a novel scaling regime where both the server needs of jobs and the system load scale with the total number of servers in the system. Our main result is an upper bound on the queueing probability under FCFS scheduling, which allows us to establish conditions for the queueing probability to go to zero as the number of servers grows. We also characterize the transient behavior of the system under constant load in the large-system limit.

There are many interesting directions that are worth further investigation in future work. Here we list a few.

(1) As we mentioned in Sections 1 and 2, finding exact (non-asymptotic) stability conditions under FCFS scheduling for more than two job classes with heterogeneous service rates is open.
(2) Characterizing the job response time in non-asymptotic regimes is wide open.
(3) It is of great interest to extend our stability condition and queueing probability upper bound to settings with service time distributions beyond exponential distributions. For this direction, a natural first attempt would be to generalize the Lyapunov drift-based analysis in this paper. A recent work [50] has demonstrated that the drift method can be used to obtain tight bounds in heavy traffic under phase-type service time distributions. However, the analysis in [50] requires a complicated construction of the Lyapunov function. We anticipate that finding a proper Lyapunov function for the multi-server job model will be highly challenging.

(4) Network structure is playing an increasingly important role in job scheduling due to data locality [51, 57], which constrains the servers on which a job can run. Recent works [37, 42, 53] have studied how the network structure affects performance in load-balancing systems. Analyzing the multi-server job model with a network structure is an interesting future direction. Here the network structure can be modeled as follows: a multi-server job not only specifies how many servers it needs, but also which servers it can run on. Then a fundamental question is: What are the conditions on job server needs, system load, and additionally, *network topology*, that result in zero queueing?

ACKNOWLEDGMENTS.

We thank Sem Borst for his insightful comments on the paper. This work was supported in part by NSF grants CIF-1409106, CMMI-1938909, XPS-1629444, CSR-1763701, CNS-2007733 and CNS-1955997; and by a Google 2020 Faculty Research Award.

REFERENCES

[1] 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)*. Savannah, GA, 265–283.

[2] Larisa Afanaseva, Elena Bashtova, and Svetlana Grishunina. 2019. Stability analysis of a multi-server model with simultaneous service and a regenerative input flow. *Methodology and Computing in Applied Probability* (2019), 1–17.

[3] 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)*. 547–564.

[4] François Baccelli and Serguei Foss. 1995. On the Saturation Rule for the Stability of Queues. *J. Appl. Probab.* 32, 2 (1995), 494–507.

[5] 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.

[6] N. Benameur, S. Ben Fredj, F. Delcoigne, S. Oueslati-Boulahia, and J.W. Roberts. 2001. Integrated Admission Control for Streaming and Elastic Traffic. In *Int. Workshop Quality of Future Internet Services (QofIS)*. 69–81.

[7] Dimitris Bertsimas, David Gamarnik, and John N. Tsitsiklis. 2001. Performance of Multiclass Markovian Queueing Networks Via Piecewise Linear Lyapunov Functions. *Ann. Appl. Probab.* 11, 4 (11 2001), 1384–1428.

[8] Thomas Bonald and Céline Comte. 2017. Balanced fair resource sharing in computer clusters. *Perform. Eval.* 116 (2017), 70 – 83.

[9] 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.

[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] A. Dasylva and R. Srikant. 1999. Bounds on the Performance of Admission Control and Routing Policies for General Topology Networks with Multiple Call Centers. In *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, Vol. 2. New York, NY, 505–512.

[12] Moez Draief and Laurent Massoulié. 2009. *Epidemics and Rumours in Complex Networks.* Cambridge University Press.

[13] 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.

[14] D. Filippopoulos and H. Karatza. 2007. An M/M/2 parallel system model with pure space sharing among rigid jobs. *Mathematical and Computer Modelling* 45, 5 (2007), 491–530.

[15] Isaac Grosof, Mor Harchol-Balter, and Alan Scheller-Wolf. 2020. Stability for Two-class Multiserver-job Systems. arXiv:2010.00631.

[16] Bruce Hajek. 1982. Hitting-Time and Occupation-Time Bounds Implied by Drift Analysis with Applications. *Adv. Appl. Probab.* 14, 3 (1982), 502–525.

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

[18] P. J. Hunt and T. G. Kurtz. 1994. Large loss networks. *Stoch. Proc. Appl.* 53, 2 (1994), 363 – 378.

[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] Donald L. Iglehart. 1973. Weak convergence of compound stochastic process, I. *Stoch. Proc. Appl.* 1, 1 (1973), 11 – 31.

[21] Sung Shick Kim. 1979. *M/M/s queueing system where customers demand multiple server use.* Ph.D. Dissertation. Southern Methodist University.

[22] A. E. Krzesinski. 2011. *Order Independent Queues.* Springer US, Boston, MA, 85–120.

[23] Thomas G. Kurtz. 1981. *Approximation of Population Processes.* Society for Industrial and Applied Mathematics.

[24] 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.

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

[26] Xin Liu, Kang Gong, and Lei Ying. 2020. Steady-State Analysis of Load Balancing with Coxian-2 Distributed Service Times. *arXiv:2005.09815 [math.PR]* (2020).

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

[28] 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.

[29] Yi Lu, Qiaomin Xie, Gabriel Kliot, Alan Geller, James R. Larus, and Albert Greenberg. 2011. Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services. *Perform. Eval.* 68, 11 (Nov. 2011), 1056–1071.

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

[31] 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.

[32] 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.

[33] Agassi Melikov. 1996. Computation and Optimization Methods for Multiresource Queues. *Cybern. Syst. Anal.* 32, 6 (1996), 821–836.

[34] A. Z. Melikov. 1996. Computation and Optimization Methods for Multiresource Queues. *Cybernetics and Systems Analysis* 32, 6 (1996).

[35] Micheal David Mitzenmacher. 1996. *The Power of Two Choices in Randomized Load Balancing.* Ph.D. Dissertation. University of California at Berkeley.

[36] 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.

[37] Debankur Mukherjee, Sem C. Borst, and Johan S.H. van Leeuwaarden. 2018. Asymptotically Optimal Load Balancing Topologies. *Proc. ACM SIGMETRICS Int. Conf. Measurement and Modeling of Computer Systems* 2, 1, Article 14 (April 2018), 29 pages.

[38] Leonid Ponomarenko, Che Soong Kim, and Agassi Melikov. 2010. *Performance analysis and optimization of multi-traffic on communication networks.* Springer Science & Business Media.

[39] 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.

[40] 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.

[41] Alexander Rumyantsev and Evsey Morozov. 2017. Stability criterion of a multiserver model with simultaneous service. *Annals of Operations Research* 252, 1 (2017), 29–39.

[42] Daan Rutten and Debankur Mukherjee. 2021. Load balancing under strict compatibility constraints. In *Proc. ACM SIGMETRICS Int. Conf. Measurement and Modeling of Computer Systems.*

[43] R. Srikant and Lei Ying. 2014. *Communication Networks: An Optimization, Control and Stochastic Networks Perspective.* Cambridge Univ. Press, New York.

[44] Alexander L. Stolyar. 2015. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Syst.* 80, 4 (Aug. 2015), 341–361.

[45] Oleg M. Tikhonenko. 2005. Generalized Erlang Problem for Service Systems with Finite Total Capacity. *Problems of Information Transmission* 41, 3 (2005), 243–253.

[46] 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.

[47] Nico M. van Dijk. 1989. Blocking of Finite Source Inputs Which Require Simultaneous Servers with General Think and Holding Times. *Operations Research Letters* 8, 1 (February 1989), 45 – 52.
[48] 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)*. ACM, 18.
[49] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich. 1996. Queueing System with Selection of the Shortest of Two Queues: An Asymptotic Approach. *Probl. Inf. Transm.* 32, 1 (1996), 15–27.
[50] 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. *ACM SIGMETRICS Perform. Evaluation Rev.* 45, 3 (March 2018), 232–245.
[51] Weina Wang, Kai Zhu, Lei Ying, Jian Tan, and Li Zhang. 2013. A throughput optimal algorithm for map task scheduling in MapReduce with data locality. *ACM SIGMETRICS Perform. Evaluation Rev.* 40, 4 (March 2013), 33–42.
[52] Wentao Weng and Weina Wang. 2021. Achieving Zero Asymptotic Queueing Delay for Parallel Jobs. In *Proc. ACM SIGMETRICS Int. Conf. Measurement and Modeling of Computer Systems*.
[53] Wentao Weng, Xinyu Zhou, and R. Srikant. 2021. Optimal Load Balancing with Locality Constraints. In *Proc. ACM SIGMETRICS Int. Conf. Measurement and Modeling of Computer Systems*.
[54] Ward Whitt. 1985. Blocking when service is required from several facilities simultaneously. *AT&T Tech. J.* 64 (1985), 1807 – 1856.
[55] John Wilkes. 2019. Google cluster-usage traces v3. http://github.com/google/cluster-data.
[56] 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.
[57] Qiaomin Xie and Yi Lu. 2015. Priority algorithm for near-data scheduling: Throughput and heavy-traffic optimality. In *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*. Hong Kong, China, 963–972.

A PROOF OF THEOREM 4.3

We first prove **Step 1**. Consider the queue length process $\{Y_i^{(n)}(t): t \geq 0\}$ for class i. It is clear that the queue length increases by 1 with rate $\lambda_i^{(n)}$ and decreases by 1 with rate $\mu_i \cdot \min\left\{Y_i^{(n)}(t), \frac{n}{m_i^{(n)}}\right\}$. Let $\{N_{i,1}(t): t \geq 0\}$ and $\{N_{i,2}(t): t \geq 0\}$ for $i = 1, 2, \ldots, K$ be independent unit-rate Poisson processes. Then $Y_i^{(n)}(t)$ can be constructed as follows:

$$Y_i^{(n)}(t) = Y_i^{(n)}(0) + N_{i,1}\left(\int_0^t \lambda_i^{(n)} ds\right) - N_{i,2}\left(\int_0^t \mu_i \min\left\{Y_i^{(n)}(s), \frac{n}{m_i^{(n)}}\right\} ds\right).$$

We consider a scaled version of $Y_i^{(n)}(t)$, defined as $\widetilde{Y}_i^{(n)} = \frac{1}{\lambda_i^{(n)}} Y_i^{(n)}(t)$. We have

$$\begin{aligned}
\widetilde{Y}_i^{(n)}(t) =& \widetilde{Y}_i^{(n)}(0) + \frac{1}{\lambda_i^{(n)}} N_{i,1}\left(\int_0^t \lambda_i^{(n)} ds\right) - \frac{1}{\lambda_i^{(n)}} N_{i,2}\left(\int_0^t \lambda_i^{(n)} \mu_i \min\left\{\widetilde{Y}_i^{(n)}(s), \frac{n}{\lambda_i^{(n)} m_i^{(n)}}\right\} ds\right) \\
=& \widetilde{Y}_i^{(n)}(0) + \frac{1}{\lambda_i^{(n)}} \overline{N}_{i,1}\left(\int_0^t \lambda_i^{(n)} ds\right) - \frac{1}{\lambda_i^{(n)}} \overline{N}_{i,2}\left(\int_0^t \lambda_i^{(n)} \mu_i \min\left\{\widetilde{Y}_i^{(n)}(s), \frac{n}{\lambda_i^{(n)} m_i^{(n)}}\right\} ds\right) \\
&+ \int_0^t \left(1 - \mu_i \min\left\{\widetilde{Y}_i^{(n)}(s), \frac{n}{\lambda_i^{(n)} m_i^{(n)}}\right\}\right) ds, \qquad (22)
\end{aligned}$$

where $\overline{N}_{i,1}(t) = N_{i,1}(t) - t$ and $\overline{N}_{i,2}(t) = N_{i,2}(t) - t$ are the centered Poisson processes. Consider the term (22) and note that $\frac{n\mu_i}{\lambda_i^{(n)} m_i^{(n)}} = \frac{1}{\rho_i}$. Define a function $F_i: \mathbb{R}_+ \to \mathbb{R}$ by $F_i(x) = 1 - \min\{\mu_i x, 1/\rho_i\}$. Then (22) can be written as $\int_0^t F_i\left(\widetilde{Y}_i^{(n)}(s)\right) ds$. It can be easily verified that $F_i(\cdot)$ is μ_i-Lipschitz. Note that the fluid model $y_i(t)$ can be written in an integral form as follows:

$$y_i(t) = y_i(0) + \int_0^t \left(1 - \mu_i \min\left\{\mu_i y_i(t), \frac{1}{\rho_i}\right\}\right) ds = y_i(0) + \int_0^t F_i\left(y_i(s)\right) ds.$$

Now we compare $\widetilde{Y}_i^{(n)}(t)$ with $y_i(t)$:

$$\begin{aligned}
&\left|\widetilde{Y}_i^{(n)}(t) - y_i(t)\right| \\
\le &\left|\widetilde{Y}_i^{(n)}(0) - y_i(0)\right| + \int_0^t \left|F_i\left(\widetilde{Y}_i^{(n)}(s)\right) - F_i\left(y_i^{(n)}(t)\right)\right| ds \\
&+ \frac{1}{\lambda_i^{(n)}}\left|\overline{N}_{i,1}\left(\int_0^t \lambda_i^{(n)} ds\right)\right| + \frac{1}{\lambda_i^{(n)}}\left|\overline{N}_{i,2}\left(\int_0^t \lambda_i^{(n)}\mu_i \min\left\{\widetilde{Y}_i^{(n)}(s), \frac{n}{\lambda_i^{(n)} m_i^{(n)}}\right\} ds\right)\right| \\
\le &\left|\widetilde{Y}_i^{(n)}(0) - y_i(0)\right| + \int_0^t \mu_i \left|\widetilde{Y}_i^{(n)}(s) - y_i^{(n)}(t)\right| ds \\
&+ \frac{1}{\lambda_i^{(n)}}\left|\overline{N}_{i,1}\left(\int_0^t \lambda_i^{(n)} ds\right)\right| + \frac{1}{\lambda_i^{(n)}}\left|\overline{N}_{i,2}\left(\int_0^t \lambda_i^{(n)}\mu_i \min\left\{\widetilde{Y}_i^{(n)}(s), \frac{n}{\lambda_i^{(n)} m_i^{(n)}}\right\} ds\right)\right|, \qquad (23)
\end{aligned}$$

where the second inequality follows from the Lipschitz property of $F_i(\cdot)$.

Next we bound the terms in (23). By applying a classical inequality on unit-rate Poisson process (see Proposition 5.2 in [12]), we have

$$\mathbb{P}\left(\sup_{0\le t\le T}\frac{1}{\lambda_i^{(n)}}\left|\overline{N}_{i,1}\left(\int_0^t \lambda_i^{(n)} ds\right)\right| > \epsilon'\right) = \mathbb{P}\left(\sup_{0\le t\le \lambda_i^{(n)} T}\left|\overline{N}_{i,1}(t)\right| > \epsilon'\lambda_i^{(n)}\right) \le 2e^{-\lambda_i^{(n)} T\cdot h\left(\frac{\epsilon'}{T}\right)},$$

where h is a function defined by $h(u) = (1+u)\log(1+u) - u$. Similarly,

$$\begin{aligned}
&\mathbb{P}\left(\sup_{0\le t\le T}\frac{1}{\lambda_i^{(n)}}\left|\overline{N}_{i,2}\left(\int_0^t \lambda_i^{(n)}\mu_i \min\left\{\widetilde{Y}_i^{(n)}(s), \frac{n}{\lambda_i^{(n)} m_i^{(n)}}\right\} ds\right)\right| > \epsilon'\right) \\
\le &\mathbb{P}\left(\sup_{0\le t\le n\mu_i T/m_i^{(n)}}\left|\overline{N}_{i,2}(t)\right| > \epsilon'\lambda_i^{(n)}\right) \le 2e^{-\frac{n\mu_i T}{m_i^{(n)}} h\left(\frac{\rho_i\epsilon'}{T}\right)}.
\end{aligned}$$

Combining these bounds, we have

$$\begin{aligned}
&\mathbb{P}\left(\sup_{0\le t\le T}\left(\left|\widetilde{Y}_i^{(n)}(t) - y_i(t)\right| - \int_0^t \mu_i\left|\widetilde{Y}_i^{(n)}(s) - y_i(t)\right| ds\right) > 3\epsilon'\right) \\
\le &\mathbb{P}\left(\left|\widetilde{Y}_i^{(n)}(0) - y_i(0)\right| > \epsilon'\right) + 2e^{-\lambda_i^{(n)} T\cdot h\left(\frac{\epsilon'}{T}\right)} + 2e^{-\frac{n\mu_i T}{m_i^{(n)}} h\left(\frac{\rho_i\epsilon'}{T}\right)}.
\end{aligned}$$

Note that the function $\left|\widetilde{Y}_i^{(n)}(t) - y_i(t)\right|$ is finite with probability 1 on the interval $[0,T]$. Applying Gronwall's lemma yields

$$\begin{aligned}
&\mathbb{P}\left(\sup_{0\le t\le T}\left|\widetilde{Y}_i^{(n)}(t) - y_i(t)\right| > \frac{\epsilon}{K}\right) \\
\le &\mathbb{P}\left(\sup_{0\le t\le T}\left(\left|\widetilde{Y}_i^{(n)}(t) - y_i(t)\right| - \int_0^t \mu_i\left|\widetilde{Y}_i^{(n)}(s) - y_i(t)\right| ds\right) > \frac{\epsilon e^{-\mu_i T}}{K}\right) \\
\le &\mathbb{P}\left(\left|\widetilde{Y}_i^{(n)}(0) - y_i(0)\right| > \frac{\epsilon e^{-\mu_i T}}{3K}\right) + 2e^{-\lambda_i^{(n)} T\cdot h\left(\frac{\epsilon e^{-\mu_i T}}{3KT}\right)} + 2e^{-\frac{n\mu_i T}{m_i^{(n)}} h\left(\frac{\rho_i\epsilon e^{-\mu_i T}}{3KT}\right)}, \qquad (24)
\end{aligned}$$

where the last step is obtained by setting $\epsilon' = \frac{\epsilon e^{-\mu_i T}}{3K}$. This completes Step 1.

Next we prove **Step 2**. As we noted, $X_i^{(n)}(t) = Y_i^{(n)}(t)$ for all class i and all $t \in [0,T]$ if $\sup_{0\le t\le T}\sum_{i=1}^K m_i^{(n)} Y_i^{(n)}(t) \le n$. We make the following claim (the proof is provided at the end).

Claim 1. *The inequality* $\sup_{0\le t\le T}\sum_{i=1}^K m_i^{(n)}Y_i^{(n)}(t)\le n$ *holds true if, for all* i,

$$\sup_{0\le t\le T}\left|\widetilde{Y}_i^{(n)}(t)-y_i(t)\right|\le\delta, \tag{25}$$

where $\delta=\frac{1}{\mu_{\max}\rho}\left(1-\sum_{i:y_i(0)\ge 1/\mu_i}\rho_i\mu_i y_i(0)-\sum_{i:y_i(0)<1/\mu_i}\rho_i\right)$ *is a positive constant.*

By Claim 1, we have

$$\begin{aligned}
&\mathbb{P}\left(\exists\, i \text{ and } t\in[0,T] \text{ s.t. } X_i^{(n)}(t)\ne Y_i^{(n)}(t)\right)\\
\le&\sum_{i=1}^K\mathbb{P}\left(\sup_{0\le t\le T}\left|\widetilde{Y}_i^{(n)}(t)-y_i(t)\right|>\delta\right)\\
\le&\sum_{i=1}^K\left(\mathbb{P}\left(\left|\widetilde{Y}_i^{(n)}(0)-y_i(0)\right|>\frac{\delta}{3}\right)+2e^{-\lambda_i^{(n)}T\cdot h\left(\frac{\delta e^{-\mu_i T}}{3T}\right)}+2e^{-\frac{n\mu_i T}{m_i^{(n)}}h\left(\frac{\delta\rho_i e^{-\mu_i T}}{3T}\right)}\right).
\end{aligned}\tag{26}$$

As the last step, we plug the upper bounds (24) and (26) into (18) and (17) and obtain:

$$\begin{aligned}
&\mathbb{P}\left(\sup_{0\le t\le T}\sum_{i=1}^K\left|\widetilde{X}_i^{(n)}-y_i(t)\right|>\epsilon\right)\\
\le&2\sum_{i=1}^K\left(e^{-\lambda_i^{(n)}T\cdot h\left(\frac{\epsilon e^{-\mu_i T}}{3KT}\right)}+e^{-\frac{n\mu_i T}{m_i^{(n)}}h\left(\frac{\rho_i\epsilon e^{-\mu_i T}}{3KT}\right)}+e^{-\lambda_i^{(n)}T\cdot h\left(\frac{\delta e^{-\mu_i T}}{3T}\right)}+e^{-\frac{n\mu_i T}{m_i^{(n)}}h\left(\frac{\rho_i\delta e^{-\mu_i T}}{3T}\right)}\right) &(27)\\
&+\sum_{i=1}^K\left(\mathbb{P}\left(\left|\widetilde{Y}_i^{(n)}(0)-y_i(0)\right|>\frac{\epsilon e^{-\mu_i T}}{3K}\right)+\mathbb{P}\left(\left|\widetilde{Y}_i^{(n)}(0)-y_i(0)\right|>\frac{\delta}{3}\right)\right) &(28)
\end{aligned}$$

By the assumption that $\widetilde{Y}_i^{(n)}(0)=\widetilde{X}_i^{(n)}(0)$ and $\lim_{n\to\infty}\widetilde{X}_i^{(n)}(0)=y_i(0)$ in probability, we have

$$\lim_{n\to\infty}\mathbb{P}\left(\left|\widetilde{Y}_i^{(n)}(0)-y_i(0)\right|>\frac{\epsilon e^{-\mu_i T}}{3K}\right)=0,\qquad\lim_{n\to\infty}\mathbb{P}\left(\left|\widetilde{Y}_i^{(n)}(0)-y_i(0)\right|>\frac{\delta}{3}\right)=0.$$

With $m_{\max}^{(n)}=o(n)$, it follows that $\lambda_i^{(n)}=\frac{n\rho_i\mu_i}{m_i^{(n)}}\to\infty$ as $n\to\infty$, thus the terms in (27) converge to 0 as $n\to\infty$. Therefore,

$$\lim_{n\to\infty}\mathbb{P}\left(\sup_{0\le t\le T}\sum_{i=1}^K\left|\widetilde{X}_i^{(n)}-y_i(t)\right|>\epsilon\right)=0,\qquad\forall\epsilon>0.$$

This completes the proof of Theorem 4.3. □

Proof of Claim 1. By the condition in (25), we have

$$\sum_{i=1}^K m_i^{(n)}Y_i^{(n)}(t)=\sum_{i=1}^K n\rho_i\mu_i\widetilde{Y}_i^{(n)}(t)\le n\delta\mu_{\max}\rho+n\sum_{i=1}^K\rho_i\mu_i y_i(t).$$

Recall that the initial condition $\boldsymbol{y}(0)$ satisfies $y_i(0)<\frac{1}{\rho\mu_i}<\frac{1}{\rho_i\mu_i}$ for all i. We then have $y_i(t)=\left(y_i(0)-\frac{1}{\mu_i}\right)e^{-\mu_i t}+\frac{1}{\mu_i}$, for each $i\in\{1,\dots,K\}$. Therefore, for any $t\in[0,T]$,

$$\sum_{i=1}^K\rho_i\mu_i y_i(t)=\sum_{i=1}^K\rho_i\mu_i\left[\left(y_i(0)-\frac{1}{\mu_i}\right)e^{-\mu_i t}+\frac{1}{\mu_i}\right]\le\sum_{i:y_i(0)\ge 1/\mu_i}\rho_i\mu_i y_i(0)+\sum_{i:y_i(0)<1/\mu_i}\rho_i\triangleq\alpha.$$

Note that $\alpha < \sum_{i:y_i(0)\geq 1/\mu_i} \frac{\rho_i}{\rho} + \sum_{i:y_i(0)<1/\mu_i} \frac{\rho_i}{\rho} = 1$. With $\delta = \frac{1-\alpha}{\mu_{\max}\rho}$, it follows that

$$\sup_{0\leq t\leq T} \sum_{i=1}^{K} m_i^{(n)} Y_i^{(n)}(t) \leq n\delta\mu_{\max}\rho + n\alpha = n.$$

This completes the proof of Claim 1. □

Received October 2020; revised December 2020; accepted January 2021