

Achieving Zero Asymptotic Queuing Delay for Parallel Jobs

WENTAO WENG, Institute for Interdisciplinary Information Sciences, Tsinghua University, China
WEINA WANG, Computer Science Department, Carnegie Mellon University, USA

Zero queuing delay is highly desirable in large-scale computing systems. Existing work has shown that it can be asymptotically achieved by using the celebrated Power-of- d -choices (Pod) policy with a probe overhead $d = \omega\left(\frac{\log N}{1-\lambda}\right)$, and it is impossible when $d = O\left(\frac{1}{1-\lambda}\right)$, where N is the number of servers and λ is the load of the system. However, these results are based on the model where each job is an *indivisible* unit, which does not capture the parallel structure of jobs in today's predominant parallel computing paradigm.

This paper thus considers a model where each job consists of a batch of parallel tasks. Under this model, we propose a new notion of zero (asymptotic) queuing delay that requires the job delay under a policy to approach the job delay given by the max of its tasks' service times, i.e., the job delay assuming its tasks entered service right upon arrival. This notion quantifies the effect of queuing on a *job level* for jobs consisting of multiple tasks, and thus deviates from the conventional zero queuing delay for single-task jobs in the literature.

We show that zero queuing delay for parallel jobs can be achieved using the *batch-filling policy* (a variant of the celebrated Pod policy) with a probe overhead $d = \omega\left(\frac{1}{(1-\lambda)\log k}\right)$ in the sub-Halfin-Whitt heavy-traffic regime, where k is the number of tasks in each job and k properly scales with N (the number of servers). This result demonstrates that for *parallel jobs*, zero queuing delay can be achieved with a smaller probe overhead. We also establish an impossibility result: we show that zero queuing delay cannot be achieved if $d = \exp\left(o\left(\frac{\log N}{\log k}\right)\right)$. Simulation results are provided to demonstrate the consistency between numerical results and theoretical results under reasonable settings, and to investigate gaps in the theoretical analysis.

CCS Concepts: • **Mathematics of computing** → **Queueing theory; Markov processes**; • **Networks** → **Network performance analysis**; • **Theory of computation** → **Parallel computing models**; • **Computer systems organization** → **Cloud computing**.

Additional Key Words and Phrases: parallel computing; load balancing; power-of-d-choices; delay performance

ACM Reference Format:

Wentao Weng and Weina Wang. 2020. Achieving Zero Asymptotic Queuing Delay for Parallel Jobs. *Proc. ACM Meas. Anal. Comput. Syst.* 4, 3, Article 42 (December 2020), 36 pages. <https://doi.org/10.1145/3428327>

1 INTRODUCTION

In view of the rise in the amount of latency-critical workloads in today's datacenters [31, 37], load-balancing policies with ultra-low latency have attracted great attention (see, e.g., [12, 23–25, 28]). In particular, it is highly desirable to have a policy under which the delay due to queueing is minimal.

In a classical setting of load-balancing, the celebrated greedy policy, Join-the-Shortest-Queue (JSQ), achieves a minimal queuing delay in the sense that the queuing delay is *diminishing* as

Authors' addresses: Wentao Weng, wwt17@mails.tsinghua.edu.cn, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, 100084, China; Weina Wang, weinaw@cs.cmu.edu, Computer Science Department, Carnegie Mellon University, PA, 15213-3890, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

2476-1249/2020/12-ART42 \$15.00

<https://doi.org/10.1145/3428327>

the system becomes large, even in heavy-traffic regimes [28, 41, 42]. Therefore, we say that JSQ achieves a *zero (asymptotic) queueing delay*. Specifically, consider a system with N servers where jobs arrive into the system following a Poisson process. Each server has its own queue and serves jobs in the queue in a First-Come-First-Serve manner. Under JSQ, each incoming job will be assigned to a server with the shortest queue length. Then the expected time (in steady state) a job spends in the queue *before* entering service goes to zero as N goes to infinity.

However, a drawback of JSQ is that it has a high communication overhead, which can cancel out its advantage of achieving zero queueing delay. For assigning each job, JSQ requires the knowledge of the queue-length information of all the N servers, which will be referred to as having a *probe overhead* of N . In a typical cluster of servers, N is in the tens of thousands range, resulting in intolerable delay due to communication [31, 37].

A load-balancing algorithm that provides tradeoffs between queueing delay and communication overhead is the Power-of- d -choices (Pod) policy [27, 38]. For each incoming job, Pod selects d queues out of N queues uniformly at random, and assigns the job to a shortest queue among the d selected queues. Therefore, Pod has a probe overhead of d . It is easy to see that when $d = N$, Pod coincides with JSQ, thus achieving a zero queueing delay. However, a fundamental question is: *Can zero queueing delay be achieved by Pod with a d value smaller than N ? Or, what is the smallest d for achieving zero queueing delay?*

This question has been recently answered in a line of research [23–25, 28]. In particular, the following results are the most relevant to our paper. Suppose the job arrival rate is $N\lambda$ and job service times are exponentially distributed with rate 1. Then the load of the system is λ . Consider a heavy-traffic regime with $\lambda = 1 - \beta N^{-\alpha}$, where α and β are constants with $0 < \beta \leq 1$ and $0 < \alpha < 1$. It has been shown that Pod achieves zero queueing delay when $d = \Omega\left(\frac{\log N}{1-\lambda}\right)$ for $\alpha \in (0, 0.5)$ and when $d = \Omega\left(\frac{\log^2 N}{1-\lambda}\right)$ for $\alpha \in [0.5, 1)$; and it does not have zero queueing delay when $d = O\left(\frac{1}{1-\lambda}\right)$. However, although these prior results provide great insights into achieving zero queueing delay, they are all for the classical setting where each job is an indivisible unit.

In today’s applications, parallel computing is becoming increasingly popular to support the rapidly growing data volume and computation demands, especially in large scale clusters that support data-parallel frameworks such as [36, 46]. A job with a parallel structure is no longer a single unit, but rather has multiple components that can run in parallel. In particular, the vast number of data analytic and scientific computing workloads are parallel or embarrassingly parallel [21, 30, 31]. Additional application examples include data replications in distributed file systems [8, 22] and hyper-parameter tuning and Monte-Carlo search in machine learning [21, 29].

In this paper, inspired by this emerging paradigm of parallel computing, we revisit the fundamental question on the minimum probe overhead needed for achieving zero queueing delay, and answer it under parallelism. To capture the parallel structure, we consider a model where each job consists of k tasks that can run on different servers in parallel. We assume that task service times are independent and exponentially distributed with rate 1. Under such a model, we focus on delay performance on a job level, i.e., we are interested in *job delay*, which is the time from when a job arrives until all of its tasks are completed. We choose this performance metric since usually a job is a meaningful unit for users. In fact, minimizing the delay of jobs, rather than the delay of their tasks, is the design goal of many practical schedulers [4, 9, 16, 31].

We reiterate that we consider the asymptotic regime that $N \rightarrow \infty$. We assume that k , the number of tasks per job, properly scales with N .

Zero queueing delay for parallel jobs. The term “zero queueing delay” is usually used to refer to the regime where the delay due to queueing is minimal, i.e., where jobs barely wait behind each

other and are thus only subject to delay due to their inherent sizes. In the non-parallel model, it is clear that the delay due to queueing for a job is just the time a job spends waiting in the queue. However, when a job consists of *multiple* tasks, quantifying the delay due to queueing is more complicated since different tasks experience different queueing times.

In this paper, we propose the following notion of zero queueing delay for parallel jobs. Let X_1, X_2, \dots, X_k denote the service times of a job's k tasks. Then if a job does not experience any queueing, its delay is given by $T^* = \max\{X_1, X_2, \dots, X_k\}$. This is the job delay when all the tasks of the job enter service immediately, so we call it the *inherent delay*. Note that here the inherent delay is *not* the total size of all the tasks of a job, but rather the delay of the job when it is parallelized. Let T denote the delay of a job in steady state under a load-balancing policy. Then the delay due to queueing can be characterized by the difference $\mathbb{E}[T - T^*]$. We say zero queueing delay is achieved if

$$\frac{\mathbb{E}[T - T^*]}{\mathbb{E}[T^*]} \rightarrow 0 \quad \text{as } N \rightarrow \infty, \quad (1)$$

i.e., the queueing delay takes a diminishing fraction of the inherent delay.

Our notion of zero queueing delay recovers the conventional notion for non-parallel jobs when $k = 1$. However, it is different from the requirement that under the parallel job model, all the tasks of a job should have zero queueing delay. Such a requirement is rather strong since all the tasks would need to be assigned to empty queues *simultaneously*. We will discuss this alternative notion in more detail in Section 6.

Probe overhead and batch-filling policy. When a job arrives into the system, a task-assigning policy samples some queues to obtain their queue length information, and then decides how to assign the k tasks to the sampled servers. If the policy samples kd queues, then we say its *probe overhead* [31, 45] is d since d is the average number of samples per task.

In this paper, we focus on a policy called *batch-filling*. It samples kd queues for an incoming job and then assigns its tasks one by one to the shortest queue, where the queue length is updated after every task assignment. Batch-filling has been shown to outperform the per-task version of Pod and also a policy called batch-sampling [31, 45].

Note that the queueing dynamics under batch-filling with a probe overhead of d is also very different from that under the policy that runs Po- kd for each task, although in both policies a task gets to join the shortest queue among a set of kd queues. For this per-task version of Po- kd , tasks of the same job pick their own kd queues independently. Then it could happen that some task picks kd lightly loaded servers while some other task lands in kd highly busy servers. While under batch-filling, all the tasks in a job experience the same set of kd servers. Therefore, the analyses of batch-filling and per-task Po- kd will be very different.

Challenges and our results. The parallel structure of jobs makes a load-balancing system more challenging to analyze in the following two aspects: (i) The delay of an incoming job in steady state (tagged job) depends on the system state (queue lengths) in a more intricate way since its tasks may be assigned to different queues. (ii) The dynamics of the system state is complicated by the simultaneous arrival of a batch of tasks and the coordination in assigning tasks. Due to these intricacies, existing techniques for analyzing non-parallel models do not directly carry over to parallel models.

We address these difficulties by first deriving a sufficient condition on the state for an incoming job to achieve zero queueing delay. Notably, this condition involves all the servers whose queue lengths range from zero to a threshold that is in the order of $o(\log k)$. This is in contrast to the condition for the non-parallel model, which only depends on the fraction of idle servers. Based on this first step, we recognize that we only need to understand the system dynamics in terms of

whether the steady state concentrates around the set of desirable states that satisfy the sufficient condition. Towards this end, a key in our analysis is an interesting state-space collapse result we discover, which enables us to use the powerful framework of Stein’s method [6, 7].

Specifically, we consider a system with a job arrival rate of $N\lambda/k$. We focus on a heavy-traffic regime where the load $\lambda = 1 - \beta N^{-\alpha}$ with $0 < \beta \leq 1$ and $0 < \alpha < 0.5$, i.e., the sub-Halfin-Whitt regime. Note that the larger α is, the faster the load approaches 1 as $N \rightarrow \infty$. All the order notation and asymptotic results in this paper are with respect to the regime that $N \rightarrow \infty$.

Our main result is that zero queueing delay is *achievable* when the probe overhead d satisfies

$$d = \omega\left(\frac{1}{(1-\lambda)\log k}\right), \quad (2)$$

where the number of tasks k satisfies $k = o\left(\frac{N^{0.5-\alpha}}{\log^2 N}\right)$ and $\frac{k}{\log k} = \Omega(\log N)$. For example, this includes $k = \log^2 N$ and $k = N^{0.1}$ with $\alpha < 0.4$. Recall that for the *non-parallel* model, a lower bound result is that zero queueing *cannot* be achieved when the probe overhead is $O\left(\frac{1}{1-\lambda}\right)$. In contrast, we can see that for *parallel* jobs, the probe overhead in (2) can be orderly smaller than $\frac{1}{1-\lambda}$.

We comment that this reduction in probe overhead reflects the overall effect of parallelization on the system. There are several factors at play that are brought by parallelization all together, making it hard to quantify their individual effects. First, for tasks of the same job, the probe overhead quota is pooled together and their assignment is coordinated, leading to a more effective use of the state information. Second, a job with parallel tasks can better tolerate task delays since the job delay is anyway determined by the slowest task. Furthermore, work arrives to the system in a more bursty fashion under parallelization due to the batch effect. Such an effect of parallelization has also been investigated in some recent papers [32, 39]. But generally, understanding parallel jobs is a much underexplored research area.

To complement our achievability results, we also prove an impossibility result on the minimum probe overhead needed: zero queueing delay can not be achieved if

$$d = e^{o\left(\frac{\log N}{\log k}\right)}, \quad (3)$$

where k satisfies that $k = e^{o(\sqrt{\log N})}$ and $k = \omega(1)$. To establish this lower bound, we utilize the tail bound given by a Lyapunov function in a “reversed” way.

To the best of our knowledge, our paper is the first one that characterizes zero queueing delay on a job level for jobs with parallel tasks. The very limited amount of prior work that does study parallel jobs only has fluid-level optimality and only considers a constant load. Furthermore, we develop a new technique for lower-bounding queues, which may be of separate interest itself given the scarcity of lower-bounding techniques in queueing systems in general.

A reminder of Bachmann–Landau asymptotic notation. Since Bachmann–Landau asymptotic notation is heavily used in this paper, here we briefly recap the definitions for ease of reference. For two real-valued functions f and g of N where g takes positive values, we say that $f(N) = O(g(N))$ if there exists a positive number M such that $|f(N)| \leq M \cdot g(N)$ for large enough N , or equivalently if $\limsup_{N \rightarrow \infty} \left| \frac{f(N)}{g(N)} \right| < \infty$. We say that $f(N) = o(g(N))$ if $\lim_{N \rightarrow \infty} \frac{f(N)}{g(N)} = 0$; $f(N) = \Omega(g(N))$ if $\liminf_{N \rightarrow \infty} \frac{f(N)}{g(N)} > 0$; and $f(N) = \omega(g(N))$ if $\liminf_{N \rightarrow \infty} \left| \frac{f(N)}{g(N)} \right| = \infty$. In this paper, the asymptotic regime is when N , the number of servers, goes to infinity.

Related work. Load-balancing systems for *non-parallel* jobs have been extensively studied in the literature. It is well-known that JSQ is delay-optimal under a wide range of assumptions [41, 42].

Although getting exact-form stationary distributions is typically not feasible for most load-balancing policies, many results and approximations are known for various asymptotic regimes.

For JSQ in heavy-traffic regimes, Eschenfeldt and Gamarnik [10] obtain a diffusion approximation in the Halfin-Whitt regime ($\alpha = 0.5$), which has a zero queueing delay in the diffusion limit. The convergence result in [10] is on the process level. Braverman [5] later establish steady-state results and their results imply the convergence of the stationary distributions to the diffusion limit. JSQ has also been studied in the nondegenerate slowdown (NDS) regime ($\alpha = 1$) [17].

The problem of achieving zero queueing delay with Pod has been studied in [23–25, 28]. Mukherjee et al. [28] show through stochastic coupling that the diffusion limit of Pod with $d = \omega(N^{0.5} \log N)$ converges to that of JSQ in the Halfin-Whitt regime, thus resulting in a zero queueing delay. The convergence to the diffusion limit in [28] is on the process level. Zero queueing delay for Pod in steady state is first studied by Liu and Ying [23] for the regime where $\alpha < \frac{1}{6}$, where they show that the waiting probability goes to 0 as $N \rightarrow \infty$ when $d = \omega\left(\frac{1}{1-\lambda}\right)$. The results are later extended to the sub-Halfin-Whitt regime ($0 < \alpha < 0.5$) for both exponential and Coxian-2 service times [24, 25] and beyond-Halfin-Whitt regime ($0.5 \leq \alpha < 1$) [24], where it is shown that zero queueing delay is achieved when $d = \Omega\left(\frac{\log N}{1-\lambda}\right)$ for $\alpha \in (0, 0.5)$, and when $d = \Omega\left(\frac{\log^2 N}{1-\lambda}\right)$ for $\alpha \in [0.5, 1)$. The paper [23] also provides a lower bound result: the waiting probability is bounded away from 0 when $d = O\left(\frac{1}{1-\lambda}\right)$ for $0 \leq \alpha < 1$.

Pod has also been analyzed in the regime with a constant load ($\alpha = 0$) as $N \rightarrow \infty$. Mean-field analysis has been derived for a constant d in [27, 38], and Mukherjee et al. [28] show $d = \omega(1)$ leads to zero queueing delay. We remark that mean-field analysis results are also available for other policies such as Join-the-Idle-Queue (JIQ) [26, 34], and also for delay-resource tradeoffs [12].

To the best of our knowledge, very limited work has been done on achieving zero queueing delay for *parallel jobs*, or on analyzing delay for parallel jobs in general. Only the regime with a constant load as $N \rightarrow \infty$ has been studied. Mukherjee et al. [28] briefly touch upon this topic and show that fluid-level optimality can be achieved with probe overhead $d \geq \frac{1}{1-\lambda-\epsilon}$ under the so-called batch-sampling policy [31]. Ying et al. [45] provide limiting distributions for the stationary distributions under (batch-version) Pod, batching-sampling, and batch-filling, but have not analyzed delay of jobs. Wang et al. [39] analyze job delay under a (batch-version) random-routing policy, which does not achieve zero queueing delay. There have been no results for heavy-traffic regimes.

Finally, the techniques we use in this paper are based on Stein’s method and drift-based state-space collapse. Proposed in [33], Stein’s method has been an effective tool for bounding the distance between two distributions. The seminal papers [6, 7, 18] build an analytical framework for Stein’s method in queueing theory that consists of generator approximation, gradient bounds, and possibly state-space collapse. The papers [6, 7] use Stein’s method to study steady-state diffusion approximation, and [2, 5, 14, 15, 23, 25, 43, 44] use Stein’s method to obtain convergence rates to the mean-field limit. A similar approach has also been developed by Stolyar [35].

2 MODEL

We consider a system with N identical servers, illustrated in Figure 1. Each server has its own queue and serves tasks in its queue in a First-Come-First-Serve manner. Since each queue is associated with a server, we will refer to queues and servers interchangeably. Jobs arrive into the system following a Poisson process. To capture the parallel structure of jobs, we assume that each job consists of k tasks that can run on different servers in parallel. A job is completed when all of its tasks are completed. We study the large-system regime where the number of servers, N , becomes large, and we will let k increase to infinity with N to capture the trend of growing job sizes.

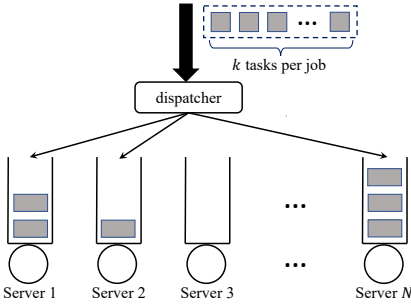


Fig. 1. A N -server system with batch arrivals.

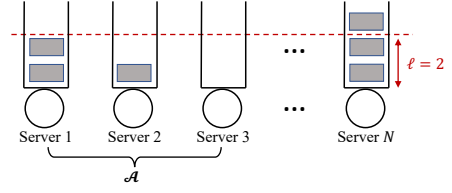


Fig. 2. An example of the number of spaces below a threshold ℓ in a set of queues: $\ell = 2$, set of queues $\mathcal{A} = \{1, 2, 3\}$, and $N_\ell(\mathcal{A}) = 3$.

We denote the job arrival rate by $N\lambda/k$ and assume that the service times of tasks are independent and exponentially distributed with rate 1. Then λ is the load of the system. We consider a heavy-traffic regime where $\lambda = 1 - \beta N^{-\alpha}$ with $0 < \beta \leq 1$ and $0 < \alpha < 0.5$, i.e., the so-called sub-Halfin-Whitt regime [19, 25].

When a job arrives into the system, we sample kd queues and obtain their queue length information. Since the average overhead is d samples per task, the probe overhead is d . We then assign the k tasks of the job to the kd selected queues using the batch-filling policy proposed in [45]. Batch-filling assigns the k tasks one by one to the shortest queue, where the queue length is updated after each task assignment. Specifically, the task assignment process runs in k rounds. For each round, we put a task into the shortest queue among sampled queues. We then update the queue length, and continue to the next round.

Now we give an equivalent description of batch-filling, which is useful in our analysis. For each queue and a positive integer ℓ , we use the *number of spaces below threshold ℓ* to refer to the quantity $\max\{\ell - \text{queue length}, 0\}$, i.e., the number of tasks we can put in the queue such that the queue length after receiving the tasks is no larger than ℓ . For a set of queues \mathcal{A} , we use $N_\ell(\mathcal{A})$ (or just N_ℓ when it is clear from the context) to denote the total number of spaces below ℓ in \mathcal{A} . Figure 2 gives an example of $N_\ell(\mathcal{A})$. We say a task is at a *queueing position p* if there are $p - 1$ tasks ahead of it in the queue. With the above terminology, the batch-filling policy can be described in the following way: it finds a minimum threshold ℓ such that the total number of spaces below ℓ in the sampled queues is at least k . Then it fills the k tasks into these spaces from low positions to high positions.

Recall that we propose the following notion of zero queueing for parallel jobs. Let X_1, X_2, \dots, X_k be the service times of the tasks of a job. If a job does not experience any queueing, its delay is given by $T^* = \max\{X_1, \dots, X_k\}$, which we call the *inherent delay* of this job. Then if the actual delay of the job is very *close* to its inherent delay, it is as if the job almost experiences no queueing. We say zero queueing delay is achieved if the steady-state job delay, T , is larger than T^* only by a diminishing fraction; i.e., if T satisfies $\mathbb{E}[T - T^*]/\mathbb{E}[T^*] \rightarrow 0$ as $N \rightarrow \infty$ as in (1). We note that as the service time of each task is exponentially distributed with mean 1, it holds that

$$\mathbb{E}[T^*] = H_k = \ln k + o(\ln k),$$

where $H_k = 1 + \frac{1}{2} + \dots + \frac{1}{k}$ is the k -th harmonic number.

We make the following interesting observation, which provides a basis for our delay analysis of parallel jobs: a job can have zero queueing delay even when its tasks are assigned to non-idle servers. In fact, we establish a necessary and sufficient condition: a job has zero queueing delay if and only if all of its tasks are at queueing positions below a threshold h with $h = o(\log k)$ after assigned to servers, noting that the inherent delay is $\ln k + o(\ln k)$. The formal proof is based on

Lemma 4.1. This phenomenon allows us to have a zero queueing delay with low probe overhead. But it also makes the analysis hard since it implies that there are many situations that can lead to zero queueing delay.

We assume that every queue has a finite buffer size of b including the task in service. If the dispatcher routes a task to a queue with length equal to b , we simply discard this task and all the other tasks of the same job. In this case, we say the job is *dropped*; otherwise, we say the job is *admitted*. We remark that this assumption is not restrictive for the following two reasons: (1) our results hold for a very large range of b (see Theorem 3.1); and (2) the probability of discarding a job is very small (see Theorem 3.2).

To represent the state of the system, let $S_i(t)$ denote the fraction of servers that have at least i jobs at time t , where $0 \leq i \leq b$. Note that it always holds $S_0(t) = 1$. Then $\mathbf{S}(t) = (S_0(t), S_1(t), \dots, S_b(t))$ forms a continuous-time Markov chain (CTMC) since batch-filling is oblivious to labels of servers. The state space is as follows:

$$\mathcal{S} = \{ \mathbf{s} = (s_0, s_1, s_2, \dots, s_b) : 1 = s_0 \geq s_1 \geq s_2 \geq \dots \geq s_b, \text{ where } Ns_i \in \mathbb{N}, \forall 1 \leq i \leq b \}.$$

It can be verified that $\{\mathbf{S}(t) : t \geq 0\}$ is irreducible and positive recurrent, thus having a unique stationary distribution. Let $\pi_{\mathcal{S}}$ denote this stationary distribution, and let $\mathbf{S} = (S_1, \dots, S_b)$ be a random element with distribution $\pi_{\mathcal{S}}$.

3 MAIN RESULTS

Our main results provide bounds on queue lengths and delay, which lead to corresponding conditions on the probe overhead for achieving zero queueing delay. We divide our results into *achievability* and *impossibility* results. Again, all the asymptotics are with respect to the regime that the number of servers, N , goes to infinity.

Achievability Results. In Theorem 3.1, we give an upper bound that characterizes $\mathbb{E} \left[\sum_{i=1}^b S_i \right]$, which is equal to the average expected number of tasks per server. This upper bound underpins our analysis of job delay.

THEOREM 3.1. Consider a system with N servers where each job consists of k tasks. Let the load be $\lambda = 1 - \beta N^{-\alpha}$ with $0 < \beta \leq 1$ and $0 < \alpha < 0.5$. Under the batch-filling policy with a probe overhead of d such that $d \geq \frac{8}{(1-\lambda)^h}$ for some $h = o(\log k)$ and $h = \omega(1)$, it holds that

$$\mathbb{E} \left[\max \left\{ \sum_{i=1}^b S_i - h \left(1 - \frac{1}{2} \beta N^{-\alpha} \right), 0 \right\} \right] \leq \frac{5}{\sqrt{N} \log N}, \quad (4)$$

where we assume that k satisfies $k = o\left(\frac{N^{0.5-\alpha}}{\log^2 N}\right)$ and $\frac{k}{\log k} = \Omega(\log N)$, the buffer size b is given by $b = \min \left\{ N^\alpha, \frac{N^{0.5-\alpha}}{k} \right\}$, and N is sufficiently large.

We remark that the $h = o(\log k)$ in this theorem represents the threshold position we pointed out for zero queueing delay, i.e., a job has zero queueing delay if all of its tasks are at queueing positions below h after assigned to servers.

The upper bound in Theorem 3.1 enables us to analyze the probability that all the tasks of an incoming job end up in positions below h under batch-filling, which further leads to the zero queueing delay result below in Theorem 3.2. Recall that the buffer size b of each queue is finite, so a job will get dropped if at least one of its tasks is assigned to a queue with a full buffer. We denote the probability of dropping an incoming job in steady state by p_d .

THEOREM 3.2. *Under the assumptions of Theorem 3.1, the steady-state delay of jobs that are admitted under batch-filling satisfies that*

$$\mathbb{E}[T \mid \text{admitted}] = \ln k + o(\ln k),$$

with a dropping probability $p_d \leq \frac{11}{b\sqrt{N}\log N}$ when N is sufficiently large.

Theorem 3.2 thus implies that zero queueing delay for parallel jobs can be achieved with a probe overhead $d = \omega\left(\frac{1}{(1-\lambda)\log k}\right)$. This breaks the lower bound of $\omega\left(\frac{1}{1-\lambda}\right)$ for achieving zero queueing delay for non-parallel jobs, i.e., single-task jobs [23], as we discussed in Section 1.

Impossibility Results. To complement the achievability results, below we investigate when zero queueing delay cannot be achieved. In Theorem 3.3, we find conditions under which $\sum_{i=1}^h S_i$ is lower bounded with a constant probability.

THEOREM 3.3. *Consider a system with N servers where each job consists of k tasks. Let the load be $\lambda = 1 - \beta N^{-\alpha}$ with $0 < \beta \leq 1$ and $0 < \alpha < 0.5$. Assume that buffers have unlimited sizes and k satisfies that $k = e^{o(\sqrt{\log N})}$ and $k = \omega(1)$. Under the batch-filling policy with a probe overhead d such that $d = e^{o\left(\frac{\log N}{\log k}\right)}$ and for any h with $h = O(\log k)$, it holds that when N is sufficiently large,*

$$\mathbb{P}\left\{\sum_{i=1}^h S_i \geq h - \frac{1}{3d}\right\} \geq \frac{1}{4e^2}. \quad (5)$$

The lower bound on $\sum_{i=1}^h S_i$ in Theorem 3.3 guarantees that an incoming job will have a significant delay in addition to its inherent delay, and thus fails to have zero queueing delay. This result is formally stated in Theorem 3.4 below.

THEOREM 3.4. *Under the assumptions of Theorem 3.3, the steady-state job delay, T , satisfies that*

$$\mathbb{E}[T] \geq 2 \ln k \quad (6)$$

when N is sufficiently large. Therefore, to achieve zero queueing delay, the probe overhead d needs to be at least $e^{\Omega\left(\frac{\log N}{\log k}\right)}$.

4 PROOFS FOR ACHIEVABILITY RESULTS (THEOREMS 3.1 AND 3.2)

Before we dive into the proofs of Theorems 3.1 and 3.2, we first develop more understanding of zero queueing delay on a job level through Lemmas 4.1 and 4.2. Due to the space limit, the proofs of the lemmas are presented in Appendix A. Then we provide a proof sketch for Theorems 3.1 and 3.2 to outline the main steps. Detailed proofs of Theorems 3.1 and 3.2 are presented in Sections 4.1 and 4.2, respectively. Throughout this section, we assume that the assumptions in Theorem 3.1 hold.

Zero queueing delay and queue lengths. Lemma 4.1 below gives an upper bound on the expected job delay given the *lengths of the queues* that the tasks of a job are assigned to. Specifically, suppose the k tasks of a job are sent to m queues ($m \leq k$) with queue lengths n_1, n_2, \dots, n_m , where the queue lengths have included these newly arrived tasks. Note that multiple tasks of the job could be sent to the same queue, but to compute the job delay, we only need to consider the last task of the job in that queue. Let Y_i with $1 \leq i \leq m$ denote the delay of the last task of the job in queue i . Then the job delay can be written as $\max\{Y_1, \dots, Y_m\}$. Lemma 4.1 gives an upper bound on $\mathbb{E}[\max\{Y_1, \dots, Y_m\}]$.

LEMMA 4.1. Consider m independent random variables Y_1, \dots, Y_m with $m \leq k$, where each Y_i ($1 \leq i \leq m$) is the sum of n_i i.i.d. random variables that follow the exponential distribution with rate 1. In the asymptotic regime that k goes to infinity, if $\max\{n_1, \dots, n_m\} = o(\log k)$, then

$$\mathbb{E}[\max\{Y_1, \dots, Y_m\}] \leq \ln k + o(\ln k).$$

The upper bound in Lemma 4.1 implies that a sufficient condition for this job to have zero queueing delay is that the lengths of the queues that its tasks are assigned to are of order $o(\log k)$. As we pointed out earlier, this is different from the single-task job model since here zero queueing delay on a job level allows non-zero queueing delay for each of the tasks.

Zero queueing delay and states. Lemma 4.2 below establishes a condition on the *state seen by a job arrival* for all of its tasks to be assigned to queues of length $o(\log k)$ with high probability, which is a sufficient condition for the job to have zero queueing delay by Lemma 4.1. Specifically, we consider the event that all the k tasks of an incoming job are assigned to queueing positions below some threshold value ℓ , and let this event be denoted by FILL_ℓ . Lemma 4.2 shows that FILL_ℓ happens with high probability given a proper condition on the state \mathbf{s} for several values of interest for ℓ . Note that if we take $\ell = h$, which is $o(\log k)$, then FILL_ℓ leads to zero queueing delay. But Lemma 4.2 is more general in the sense that it allows other values for ℓ , which is essential for other parts of the proofs including proving a state-space collapse result (Lemma 4.3) and bounding the dropping probability (Theorem 3.2).

LEMMA 4.2 (FILLING PROBABILITY). Under the assumptions of Theorem 3.1, given that the system is in a state \mathbf{s} such that

$$\sum_{i=1}^{\ell} s_i \leq \ell \left(1 - \frac{1}{4} \beta N^{-\alpha}\right), \quad (7)$$

the probability of the event FILL_ℓ for any $\ell \in \{h-1, h, b\}$ can be bounded as $\mathbb{P}\{\text{FILL}_\ell\} \geq 1 - \frac{1}{N}$ when N is sufficiently large.

Here we provide an intuitive explanation for the condition (7) when $\ell = h$. When a job arrives and sees state \mathbf{s} , if we choose one queue uniformly at random from all the queues, then the probability for the chosen queue to have a length of i is $s_i - s_{i+1}$. So the expected number of spaces below position h in the chosen queue is $\sum_{i=0}^h (h-i)(s_i - s_{i+1}) = h - \sum_{i=1}^h s_i$. The batch-filling policy samples kd queues. Thus the total expected number of spaces below position h in the kd sampled queues is $kd \left(h - \sum_{i=1}^h s_i\right)$. To fit all the k tasks of the incoming job to positions below h , we need $k \leq kd \left(h - \sum_{i=1}^h s_i\right)$, which becomes the following condition when $d \geq \frac{8}{(1-\lambda)h} = \frac{8N^\alpha}{\beta h}$ as required in Theorem 3.1:

$$\sum_{i=1}^h s_i \leq h \left(1 - \frac{1}{8} \beta N^{-\alpha}\right).$$

We strengthen this requirement to the condition $\sum_{i=1}^h s_i \leq h \left(1 - \frac{1}{4} \beta N^{-\alpha}\right)$ to obtain a high-probability guarantee using concentration bounds based on Hoeffding's results on sampling without replacement [20, Theorem 4].

Proof sketch for Theorems 3.1 and 3.2. We start by setting the goal to be proving the zero queueing delay result in Theorem 3.2, and we will see how Theorem 3.1 emerges as an essential characterization of the system that is needed for Theorem 3.2.

Considering the condition in Lemma 4.2 on the system state, we upper bound the steady-state job delay T in the following way:

$$\mathbb{E}[T] \leq \mathbb{E} \left[T \left| \sum_{i=1}^h S_i \leq h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right] \right] \quad (8)$$

$$+ \mathbb{E} \left[T \left| \sum_{i=1}^h S_i > h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right] \cdot \mathbb{P} \left\{ \sum_{i=1}^h S_i > h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right\}, \quad (9)$$

where we have used the fact that $\mathbb{P} \left\{ \sum_{i=1}^h S_i \leq h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right\} \leq 1$. We can easily bound the first summand (8) using Lemma 4.2 since this is the case where all the tasks of an incoming jobs are sent to queues with lengths no larger than h , which satisfies $h = o(\log k)$ and thus results in zero queueing delay.

We now focus on bounding the second summand (9), for which it suffices to show that the probability $\mathbb{P} \left\{ \sum_{i=1}^h S_i > h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right\}$ is small enough. By the Markov inequality,

$$\mathbb{P} \left\{ \sum_{i=1}^h S_i > h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right\} \leq \frac{\mathbb{E} \left[\max \left\{ \sum_{i=1}^b S_i - h \left(1 - \frac{1}{2} \beta N^{-\alpha} \right), 0 \right\} \right]}{\frac{1}{4} \beta N^{-\alpha}}.$$

It then boils down to bounding $\mathbb{E} \left[\max \left\{ \sum_{i=1}^b S_i - h \left(1 - \frac{1}{2} \beta N^{-\alpha} \right), 0 \right\} \right]$, which is what Theorem 3.1 achieves.

To prove Theorem 3.1, we follow the general framework of Stein's method (see, e.g., [7, 25]). The main idea is to couple our Markov chain $\{\mathbf{S}(t): t \geq 0\}$ with an auxiliary process that is easier to analyze, and bound their difference through generator approximation. In particular, we compare the dynamics of $\sum_{i=1}^b S_i(t)$ with a continuous function $x(t)$ given by the following simple fluid model as our auxiliary process:

$$\dot{x}(t) = (-\delta) \mathbb{1}_{\{x>0\}},$$

where δ is a properly chosen parameter that reflects the drift of $\sum_{i=1}^b S_i(t)$. We reiterate that a key in our analysis is a novel state-space collapse result (Lemma 4.3) that we establish, which characterizes how balanced the queues are from a job's point of view.

Combining the arguments above for bounding (8) and (9), we can conclude that the steady-state job delay $\mathbb{E}[T]$ achieves zero queueing delay.

4.1 Proof of Theorem 3.1

PROOF. As explained in the proof sketch, we compare our system with the following fluid model:

$$\dot{x}(t) = (-\delta) \mathbb{1}_{\{x>0\}}, \quad (10)$$

where $x(t)$ is continuous and $\delta = \frac{(k+1) \log N}{\sqrt{N}}$. When viewed as a continuous-time Markov chain, this fluid model (with a possibly random initial state) can be described by its generator [11], denoted as \bar{G} and given by

$$\bar{G}g(x) = g'(x) \cdot (-\delta) \mathbb{1}_{\{x>0\}}$$

for any differentiable function g . Recall that we will compare the dynamics of $\sum_{i=1}^b S_i(t)$ in our load-balancing system with $x(t)$.

The quantity of interest in Theorem 3.1 is $\mathbb{E} \left[\max \left\{ \sum_{i=1}^b S_i - \eta, 0 \right\} \right]$, where we have used the notation $\eta = h \left(1 - \frac{1}{2} \beta N^{-\alpha} \right)$ for conciseness. Recall that \mathbf{S} follows the stationary distribution of

$\{S(t): t \geq 0\}$. To couple $\{S(t): t \geq 0\}$ with the fluid model, we solve for a function g such that

$$\begin{aligned}\bar{G}g(x) &= \max\{x - \eta, 0\}, \\ g(0) &= 0.\end{aligned}\tag{11}$$

It is not hard to see that the solution is

$$g(x) = \frac{(x - \eta)^2}{2(-\delta)} \mathbb{1}_{\{x \geq \eta\}}.\tag{12}$$

Now we utilize this function g to bound $\mathbb{E}\left[\max\left\{\sum_{i=1}^b S_i - \eta, 0\right\}\right]$ through generator approximation. Let G be the generator of $\{S(t): t \geq 0\}$. Then

$$Gg\left(\sum_{i=1}^b s_i\right) = \sum_{s' \in \mathcal{S}} r_{s \rightarrow s'} \left(g\left(\sum_{i=1}^b s'_i\right) - g\left(\sum_{i=1}^b s_i\right) \right),$$

where $r_{s \rightarrow s'}$ is the transition rate from state s to s' . Since $g\left(\sum_{i=1}^b s_i\right)$ is bounded on \mathcal{S} , it holds that

$$\mathbb{E}\left[Gg\left(\sum_{i=1}^b S_i\right)\right] = 0.\tag{13}$$

Combining this with the equations in (11) gives,

$$\begin{aligned}\mathbb{E}\left[\max\left\{\sum_{i=1}^b S_i - \eta, 0\right\}\right] &= \mathbb{E}\left[\bar{G}g\left(\sum_{i=1}^b S_i\right)\right] \\ &= \mathbb{E}\left[\bar{G}g\left(\sum_{i=1}^b S_i\right) - Gg\left(\sum_{i=1}^b S_i\right)\right] \\ &= \mathbb{E}\left[g'\left(\sum_{i=1}^b S_i\right)(-\delta) - Gg\left(\sum_{i=1}^b S_i\right)\right].\end{aligned}\tag{14}$$

This is what is referred to as a *generator approximation* since we are approximating the generator G with \bar{G} .

Next we take a closer look at the term $Gg\left(\sum_{i=1}^b S_i\right)$ and derive an upper bound for (14). Let $P_A(s)$ be the probability that a job arrival is admitted into the system given that the system is at state s , i.e., the probability that all the tasks of the job are routed to positions below b . Then

$$Gg\left(\sum_{i=1}^b s_i\right) = \frac{N\lambda}{k} P_A(s) \left(g\left(\sum_{i=1}^b s_i + \frac{k}{N}\right) - g\left(\sum_{i=1}^b s_i\right) \right) + Ns_1 \left(g\left(\sum_{i=1}^b s_i - \frac{1}{N}\right) - g\left(\sum_{i=1}^b s_i\right) \right),$$

where first term is the drift due to a job arrival and the second term is due to a task departure. To derive an upper bound on (14), we divide the discussion into the three cases below. Recall that $g(x) = \frac{(x-\eta)^2}{2(-\delta)} \mathbb{1}_{\{x \geq \eta\}}$ and $g'(x) = \frac{x-\eta}{-\delta} \mathbb{1}_{\{x \geq \eta\}}$.

Case 1: $\sum_{i=1}^b S_i < \eta - \frac{k}{N}$. In this case, clearly $g'\left(\sum_{i=1}^b S_i\right) = 0$ and $Gg\left(\sum_{i=1}^b S_i\right) = 0$.

Case 2: $\sum_{i=1}^b S_i \in [\eta - \frac{k}{N}, \eta + \frac{1}{N}]$. By the mean value theorem,

$$g'\left(\sum_{i=1}^b S_i\right)(-\delta) - Gg\left(\sum_{i=1}^b S_i\right) = g'\left(\sum_{i=1}^b S_i\right)(-\delta) - \left(\frac{N\lambda}{k} P_A(S) \frac{k}{N} g'(\xi) + Ns_1 \frac{-1}{N} g'(\tilde{\xi})\right)$$

$$\leq g' \left(\sum_{i=1}^b S_i \right) (-\delta) - \lambda g'(\xi) + S_1 g'(\tilde{\xi}), \quad (15)$$

where $\xi \in \left(\sum_{i=1}^b S_i, \sum_{i=1}^b S_i + \frac{k}{N} \right)$, $\tilde{\xi} \in \left(\sum_{i=1}^b S_i - \frac{1}{N}, \sum_{i=1}^b S_i \right)$, and (15) is true since $P_A(S) \leq 1$ and $g'(x) \leq 0$ for all x .

Case 3: $\sum_{i=1}^b S_i \geq \eta + \frac{1}{N}$. Since $g'(x)$ is continuous for all x , by the second order Taylor expansion in the Lagrange form,

$$\begin{aligned} & g' \left(\sum_{i=1}^b S_i \right) (-\delta) - Gg \left(\sum_{i=1}^b S_i \right) \\ &= g' \left(\sum_{i=1}^b S_i \right) (-\delta) - \frac{N\lambda}{k} P_A(S) \left(\frac{k}{N} g' \left(\sum_{i=1}^b S_i \right) + \frac{k^2}{2N^2} g''(\zeta) \right) - NS_1 \left(\frac{-1}{N} g' \left(\sum_{i=1}^b S_i \right) + \frac{1}{2N^2} g''(\tilde{\zeta}) \right) \\ &\leq g' \left(\sum_{i=1}^b S_i \right) (-\delta - \lambda + S_1) - \frac{1}{2N} \left(\lambda k g''(\zeta) + S_1 g''(\tilde{\zeta}) \right), \end{aligned} \quad (16)$$

where $\zeta \in \left(\sum_{i=1}^b S_i, \sum_{i=1}^b S_i + \frac{k}{N} \right)$, $\tilde{\zeta} \in \left(\sum_{i=1}^b S_i - \frac{1}{N}, \sum_{i=1}^b S_i \right)$.

Combining these three cases yields

$$\begin{aligned} & \mathbb{E} \left[g' \left(\sum_{i=1}^b S_i \right) (-\delta) - Gg \left(\sum_{i=1}^b S_i \right) \right] \\ &\leq \mathbb{E} \left[\left(g' \left(\sum_{i=1}^b S_i \right) (-\delta) - \lambda g'(\xi) + S_1 g'(\tilde{\xi}) \right) \mathbb{1}_{\left\{ \sum_{i=1}^b S_i \in \left[\eta - \frac{k}{N}, \eta + \frac{1}{N} \right] \right\}} \right] \end{aligned} \quad (17)$$

$$- \frac{1}{2N} \mathbb{E} \left[\left(\lambda k g''(\zeta) + S_1 g''(\tilde{\zeta}) \right) \mathbb{1}_{\left\{ \sum_{i=1}^b S_i \geq \eta + \frac{1}{N} \right\}} \right] \quad (18)$$

$$+ \mathbb{E} \left[g' \left(\sum_{i=1}^b S_i \right) (-\delta - \lambda + S_1) \mathbb{1}_{\left\{ \sum_{i=1}^b S_i \geq \eta + \frac{1}{N} \right\}} \right]. \quad (19)$$

The first two terms (17) and (18) are easy to bound once we notice that for any $x \in \left[\eta - \frac{k+1}{N}, \eta + \frac{k+1}{N} \right]$, $|g'(x)| \leq \frac{|x-\eta|}{\delta} \leq \frac{1}{\sqrt{N} \log N}$, and for any $x \in (\eta, +\infty)$, $|g''(x)| = \frac{1}{\delta} = \frac{\sqrt{N}}{(k+1) \log N}$. Then when N is sufficiently large,

$$|(17)| \leq \frac{1}{\sqrt{N} \log N} \left(\frac{(k+1) \log N}{\sqrt{N}} + 1 + 1 \right) \leq \frac{3}{\sqrt{N} \log N},$$

and

$$|(18)| \leq \frac{1}{2N} \frac{\sqrt{N}}{(k+1) \log N} (\lambda k + 1) \leq \frac{1}{\sqrt{N} \log N}.$$

The key in this proof is to bound the term (19), for which we utilize the state-space collapse result we establish in Lemma 4.3 below. The proof of Lemma 4.3 is given in Appendix A.3.

LEMMA 4.3 (STATE-SPACE COLLAPSE). *Under the assumption of Theorem 3.1, consider the following Lyapunov function:*

$$V(\mathbf{s}) = \min \left\{ \frac{1}{h-1} \sum_{i=h}^b s_i, b \left(\left(1 - \frac{1}{2} \beta N^{-\alpha} \right) - \frac{1}{h-1} \sum_{i=1}^{h-1} s_i \right)^+ \right\},$$

where the superscript $+$ denotes the function $x^+ = \max\{x, 0\}$. Let $B = \frac{b-h+1}{h-1} \left(\beta N^{-\alpha} + \frac{\log N}{\sqrt{N}} \right)$. Then for any state \mathbf{s} such that $V(\mathbf{s}) > B$, its Lyapunov drift can be upper bounded as follows

$$\Delta V(\mathbf{s}) = GV(\mathbf{s}) \leq -\frac{b}{\sqrt{N}}.$$

Consequently, when N is sufficiently large,

$$\mathbb{P} \left\{ V(\mathbf{S}) > B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}} \right\} \leq e^{-\frac{1}{2} \log^2 N}.$$

With Lemma 4.3, we partition the probability space based on the value of $V(\mathbf{S})$ for bounding (19). Note that $g' \left(\sum_{i=1}^b S_i \right) (-\delta - \lambda + S_1) \mathbb{1}_{\{\sum_{i=1}^b S_i \geq \eta + \frac{1}{N}\}}$ is always no larger than $\frac{2b}{\delta}$ for large enough N . Then (19) can be upper bounded as:

$$(19) \leq \mathbb{E} \left[g' \left(\sum_{i=1}^b S_i \right) (-\delta - \lambda + S_1) \cdot \mathbb{1}_{\{\sum_{i=1}^b S_i \geq \eta + \frac{1}{N}\}} \left| V(\mathbf{S}) \leq B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}} \right] + \frac{2b}{\delta} \mathbb{P} \left\{ V(\mathbf{S}) > B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}} \right\}. \quad (20)$$

Now we focus on the case where we are given the condition that $V(\mathbf{S}) \leq B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}}$. Our goal is to show that S_1 is large enough such that $\delta + \lambda - S_1 < 0$. Intuitively, this condition on $V(\mathbf{S})$ implies that we either have a small $\sum_{i=h}^b S_i$, which leads to a large S_1 when combined with the condition $\sum_{i=1}^b S_i \geq \eta + \frac{1}{N}$ in the indicator, or a large $\sum_{i=1}^{h-1} S_i$, which directly gives a large S_1 since $S_1 \geq \dots \geq S_{h-1}$.

If $\frac{1}{h-1} \sum_{i=h}^b S_i \leq b \left(\left(1 - \frac{1}{2} \beta N^{-\alpha}\right) - \frac{1}{h-1} \sum_{i=1}^{h-1} S_i \right)^+$ in $V(\mathbf{S})$, the condition $V(\mathbf{S}) \leq B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}}$ implies that

$$\frac{1}{h-1} \sum_{i=h}^b S_i \leq \frac{b-h+1}{h-1} \left(\beta N^{-\alpha} + \frac{\log N}{\sqrt{N}} \right) + \frac{2kb \log^2 N}{(h-1)\sqrt{N}}. \quad (21)$$

Recall that $b = \min \left\{ N^\alpha, \frac{N^{0.5-\alpha}}{k} \right\}$ and $h = o(\log k)$. Note that the indicator function in (20) makes it sufficient to consider the case where $\sum_{i=1}^b S_i \geq \eta + \frac{1}{N}$, which implies $(h-1)S_1 + \sum_{i=h}^b S_i \geq \eta$. Combining this with (21) gives

$$\begin{aligned} S_1 &\geq \frac{\eta}{h-1} - \frac{b-h+1}{h-1} \left(\beta N^{-\alpha} + \frac{\log N}{\sqrt{N}} \right) - \frac{2kb \log^2 N}{(h-1)\sqrt{N}} \\ &\geq 1 + (1-\beta) \frac{1}{h-1} - \frac{1}{2} \beta N^{-\alpha} + o\left(\frac{1}{h}\right) \end{aligned}$$

when N is sufficiently large. Note that $\delta = o\left(\frac{1}{h}\right)$ and $\lambda = 1 - \beta N^{-\alpha}$. Therefore, $\lambda + \delta - S_1 < 0$ when N is sufficiently large.

If $\frac{1}{h-1} \sum_{i=h}^b S_i > b \left(\left(1 - \frac{1}{2} \beta N^{-\alpha}\right) - \frac{1}{h-1} \sum_{i=1}^{h-1} S_i \right)^+$ in $V(\mathbf{S})$, the condition $V(\mathbf{S}) \leq B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}}$ implies that

$$b \left(1 - \frac{1}{2} \beta N^{-\alpha} - \frac{1}{h-1} \sum_{i=1}^{h-1} S_i \right) \leq B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}}.$$

Then

$$\begin{aligned}
S_1 &\geq \frac{1}{h-1} \sum_{i=1}^{h-1} S_i \\
&\geq 1 - \frac{1}{2}\beta N^{-\alpha} - \frac{1}{b} \left(B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}} \right) \\
&\geq 1 - \frac{1}{2}\beta N^{-\alpha} + o(N^{-\alpha}).
\end{aligned}$$

As a result, again we have $\lambda + \delta - S_1 \leq -\frac{1}{2}\beta N^{-\alpha} + o(N^{-\alpha}) < 0$ when N is sufficiently large.

Inserting these bounds back to (20) gives that when N is sufficiently large,

$$\begin{aligned}
(19) &\leq 0 + \frac{2b}{\delta} \mathbb{P} \left\{ V(S) > B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}} \right\} \\
&\leq \frac{2b}{\delta} e^{-\frac{1}{2} \log^2 N} \\
&\leq \frac{1}{\sqrt{N} \log N}.
\end{aligned}$$

Combining the bounds for (17), (18) and (19), we have

$$\mathbb{E} \left[\max \left\{ \sum_{i=1}^b S_i - h \left(1 - \frac{1}{2}\beta N^{-\alpha} \right), 0 \right\} \right] \leq \frac{5}{\sqrt{N} \log N},$$

which completes the proof of Theorem 3.1. \square

4.2 Proof of Theorem 3.2

PROOF. We first bound the dropping probability p_d using Lemma 4.2 with the threshold value $\ell = b$. Note that an incoming job does not get dropped if and only if all its k tasks are routed to queueing positions below threshold b , which is the complement of the event FILL_b in Lemma 4.2. Thus,

$$\begin{aligned}
p_d &= 1 - \mathbb{P}\{\text{FILL}_b\} \\
&= 1 - \mathbb{P} \left\{ \text{FILL}_b \mid \sum_{i=1}^b S_i \leq b \left(1 - \frac{1}{4}\beta N^{-\alpha} \right) \right\} \cdot \mathbb{P} \left\{ \sum_{i=1}^b S_i \leq b \left(1 - \frac{1}{4}\beta N^{-\alpha} \right) \right\} \\
&\quad - \mathbb{P} \left\{ \text{FILL}_b \mid \sum_{i=1}^b S_i > b \left(1 - \frac{1}{4}\beta N^{-\alpha} \right) \right\} \cdot \mathbb{P} \left\{ \sum_{i=1}^b S_i > b \left(1 - \frac{1}{4}\beta N^{-\alpha} \right) \right\}.
\end{aligned}$$

We can easily have that $\mathbb{P} \left\{ \text{FILL}_b \mid \sum_{i=1}^b S_i \leq b \left(1 - \frac{1}{4}\beta N^{-\alpha} \right) \right\} \leq \frac{1}{N}$ using Lemma 4.2.

Now we bound $\mathbb{P} \left\{ \sum_{i=1}^b S_i > b \left(1 - \frac{1}{4}\beta N^{-\alpha} \right) \right\}$ using Theorem 3.1. Note that

$$\begin{aligned}
&\mathbb{P} \left\{ \sum_{i=1}^b S_i > b \left(1 - \frac{1}{4}\beta N^{-\alpha} \right) \right\} \\
&\leq \mathbb{P} \left\{ \max \left\{ \sum_{i=1}^b S_i - h \left(1 - \frac{1}{2}\beta N^{-\alpha} \right), 0 \right\} > b - \frac{b}{4}\beta N^{-\alpha} - h \right\}
\end{aligned}$$

$$\leq \mathbb{P} \left\{ \max \left\{ \sum_{i=1}^b S_i - h \left(1 - \frac{1}{2} \beta N^{-\alpha} \right), 0 \right\} > \frac{b}{2} \right\},$$

where we have used the fact that $\frac{b}{4} \beta N^{-\alpha} + h \leq \frac{b}{2}$ when N is sufficiently large due to our assumptions on b and h . Then by Markov's inequality,

$$\begin{aligned} \mathbb{P} \left\{ \sum_{i=1}^b S_i > b \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right\} &\leq \frac{\mathbb{E} \left[\max \left\{ \sum_{i=1}^b S_i - h \left(1 - \frac{1}{2} \beta N^{-\alpha} \right), 0 \right\} \right]}{\frac{b}{2}} \\ &\leq \frac{10}{b \sqrt{N} \log N}. \end{aligned}$$

Combining the arguments above yields

$$p_d \geq 1 - \frac{1}{N} - \frac{10}{b \sqrt{N} \log N} \geq 1 - \frac{11}{b \sqrt{N} \log N}$$

when N is sufficiently large.

Next we bound the expected job delay given that a job is admitted, i.e., $\mathbb{E}[T \mid \text{admitted}]$. We define the delay of a job that is dropped to be zero since it leaves the system immediately after arrival. Then $\mathbb{E}[T] = \mathbb{E}[T \mid \text{admitted}] \cdot (1 - p_d) + \mathbb{E}[T \mid \text{dropped}] \cdot p_d$, and thus $\mathbb{E}[T \mid \text{admitted}] = \frac{\mathbb{E}[T]}{1 - p_d}$. So we can focus on bounding $\mathbb{E}[T]$, following the outline given in the proof sketch.

We bound $\mathbb{E}[T]$ in the following way

$$\mathbb{E}[T] \leq \mathbb{E} \left[T \left| \sum_{i=1}^h S_i \leq h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right. \right] \quad (22)$$

$$+ \mathbb{E} \left[T \left| \sum_{i=1}^h S_i > h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right. \right] \cdot \mathbb{P} \left\{ \sum_{i=1}^h S_i > h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right\}. \quad (23)$$

For the first term (22) in this upper bound, as described in the proof sketch, we will rely on the fact that with high probability, all the k tasks are assigned to queueing positions below h . Specifically,

$$\begin{aligned} &\mathbb{E} \left[T \left| \sum_{i=1}^h S_i \leq h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right. \right] \\ &= \mathbb{E} \left[T \left| \sum_{i=1}^h S_i \leq h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right), \text{FILL}_h \right. \right] \cdot \mathbb{P} \left\{ \text{FILL}_h \left| \sum_{i=1}^h S_i \leq h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right. \right\} \\ &\quad + \mathbb{E} \left[T \left| \sum_{i=1}^h S_i \leq h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right), \overline{\text{FILL}}_h \right. \right] \cdot \mathbb{P} \left\{ \overline{\text{FILL}}_h \left| \sum_{i=1}^h S_i \leq h \left(1 - \frac{1}{4} \beta N^{-\alpha} \right) \right. \right\}, \end{aligned}$$

where $\overline{\text{FILL}}_h$ is the complement of FILL_h .

Suppose FILL_h is true. Suppose that the k tasks of the incoming job land in m distinct queues with $m \leq k$. We call the tasks with the highest positions in these m queues tasks $1, 2, \dots, m$, and let n_1, n_2, \dots, n_m denote these positions. Then the delay of task i can be written as $Y_i = \sum_{j=1}^{n_i} X_{i,j}$, where $X_{i,j}$ is the service time of the task at position j in the same queue as task i . Clearly $X_{i,j}$'s are i.i.d. with an exponential distribution of rate 1. We know that $n_i \leq h$, $i = 1, 2, \dots, m$ given FILL_h . Then by Lemma 4.1,

$$\mathbb{E}[\max \{Y_1, \dots, Y_m\}] \leq \ln k + o(\ln k).$$

When $\overline{\text{FILL}_h}$ is true, $\mathbb{E}\left[T \mid \sum_{i=1}^h S_i \leq h\left(1 - \frac{1}{4}\beta N^{-\alpha}\right), \overline{\text{FILL}_h}\right] \leq bk$ since the highest position for a task is b and the maximum is upper bounded by the sum. Further, $\mathbb{P}\left[\overline{\text{FILL}_h} \mid \sum_{i=1}^h S_i \leq h\left(1 - \frac{1}{4}\beta N^{-\alpha}\right)\right] \leq \frac{1}{N}$ by Lemma 4.2.

Combining the arguments above, we have the following bound for term (22):

$$\mathbb{E}\left[T \mid \sum_{i=1}^h S_i \leq h\left(1 - \frac{1}{4}\beta N^{-\alpha}\right)\right] \leq \ln k + o(\ln k) + \frac{bk}{N}.$$

Now we go back to the term (23). Again, it is easy to see that $\mathbb{E}\left[T \mid \sum_{i=1}^h S_i > h\left(1 - \frac{1}{4}\beta N^{-\alpha}\right)\right] \leq bk$. Utilizing Theorem 3.1, we have

$$\begin{aligned} \mathbb{P}\left\{\sum_{i=1}^h S_i > h\left(1 - \frac{1}{4}\beta N^{-\alpha}\right)\right\} &\leq \mathbb{P}\left\{\max\left\{\sum_{i=1}^b S_i - h\left(1 - \frac{1}{2}\beta N^{-\alpha}\right), 0\right\} > \frac{1}{4}h\beta N^{-\alpha}\right\} \\ &\leq \frac{\mathbb{E}\left[\max\left\{\sum_{i=1}^b S_i - h\left(1 - \frac{1}{2}\beta N^{-\alpha}\right), 0\right\}\right]}{\frac{1}{4}h\beta N^{-\alpha}} \\ &\leq \frac{20}{h\beta N^{\frac{1}{2}-\alpha} \log N}. \end{aligned}$$

With the bounds above on (22) and (23), we have

$$\mathbb{E}[T] \leq \ln k + o(\ln k) + \frac{bk}{N} + \frac{20bk}{h\beta N^{\frac{1}{2}-\alpha} \log N}.$$

Consequently,

$$\begin{aligned} \mathbb{E}[T \mid \text{admitted}] &= \frac{\mathbb{E}[T]}{1 - p_d} \\ &\leq \frac{\ln k + o(\ln k) + \frac{bk}{N} + \frac{20bk}{h\beta N^{\frac{1}{2}-\alpha} \log N}}{1 - p_d} \\ &\leq \ln k + o(\ln k), \end{aligned}$$

which completes the proof. \square

5 PROOFS FOR IMPOSSIBILITY RESULTS (THEOREMS 3.3 AND 3.4)

In this section, we prove the impossibility results in Theorems 3.3 and 3.4. Throughout this section, we assume that the assumptions in Theorem 3.3 hold true. Due to the space limit, the lemmas needed and their proofs are presented in Appendix B.

Proof sketch. We focus on proving the lower bound in Theorem 3.3 since the non-zero queuing delay result in Theorem 3.4 follows from that rather straightforwardly.

Our proof of Theorem 3.3 uses a novel lower bounding technique we develop. We derive the lower bound on $\mathbb{P}\left\{\sum_{i=1}^h S_i \geq h - \frac{1}{3d}\right\}$ by lower-bounding $\mathbb{P}\{S_1 - S_h \leq c_h\}$ for a properly chosen c_h , for which our proof proceeds in an inductive fashion.

- We first lower bound $\mathbb{P}\{S_1 - S_2 \leq c_2\}$ utilizing a tail bound for S_1 , which can be easily obtained from Little's law. This step uses Lyapunov-based tail bounds in a "reverse" way in the following sense. Typically, one can analyze the terms in the Lyapunov drift to obtain a tail bound. But

here, we utilize a tail bound obtained through other ways to bound a term (the probability in Lemma A.1) in the Lyapunov drift.

- We then lower bound $\mathbb{P}\{S_1 - S_3 \leq c_3\}$ based on the lower bound on $\mathbb{P}\{S_1 - S_2 \leq c_2\}$ in the previous step following a similar argument. We continue this procedure inductively until we get the desired lower bound on $\mathbb{P}\{S_1 - S_h \leq c_h\}$.

5.1 Proof Of Theorem 3.3

PROOF. As outlined in the proof sketch, we first lower-bound $\mathbb{P}\{S_1 - S_h \leq c_h\}$ using arguments in an inductive fashion. We start by lower-bounding $\mathbb{P}\{S_1 - S_2 \leq c_2\}$ for a properly chosen c_2 . This base case relies on the fact that $\mathbb{E}[S_1] = 1 - \beta N^{-\alpha}$, which can be easily proven using Little's law.

To simplify notation, let $u = 2kd$. Consider the Lyapunov function $V_1(\mathbf{s}) = s_1$. Let $h = O(\log k)$ and $B_1 = 1 - h\beta N^{-\alpha}$. For some state \mathbf{s} such that $V_1(\mathbf{s}) > B_1$, it holds that

$$\begin{aligned} \Delta V_1(\mathbf{s}) &= \sum_{s': \mathbf{s} \rightarrow s' \text{ due to an arrival}} r_{\mathbf{s} \rightarrow s'} (V_1(\mathbf{s}') - V_1(\mathbf{s})) + \sum_{s': \mathbf{s} \rightarrow s' \text{ due to a departure}} r_{\mathbf{s} \rightarrow s'} (V_1(\mathbf{s}') - V_1(\mathbf{s})) \\ &\stackrel{(a)}{\leq} uh\beta N^{-\alpha} - N(s_1 - s_2) \frac{1}{N} \\ &= uh\beta N^{-\alpha} - (s_1 - s_2), \end{aligned}$$

where (a) is due to Lemma B.2.

Consider the set of states $\mathcal{E}_1 = \{\mathbf{s} \in \mathcal{S} \mid s_1 - s_2 > uh^2\beta N^{-\alpha}\}$. Let $p_2 = \mathbb{P}\{\mathbf{S} \notin \mathcal{E}_1\}$, which is equal to $\mathbb{P}\{S_1 - S_2 \leq uh^2\beta N^{-\alpha}\}$. We now use the tail bound in Lemma A.1. Assume that we follow the notation in the lemma. Consider the following two cases:

- $\mathbf{s} \notin \mathcal{E}_1$, $\Delta V_1(\mathbf{s}) \leq uh\beta N^{-\alpha} =: \delta$.
- $\mathbf{s} \in \mathcal{E}_1$. Let $\gamma = -\Delta V_1(\mathbf{s})$. It holds $\gamma \geq uh\beta N^{-\alpha}(h-1)$.

Following the definition in A.1, it is easy to verify that $v_{\max} \leq \frac{k}{N}$ and $f_{\max} \leq 1$ for $V_1(\mathbf{s})$. Let $j_1 = \left(1 + \frac{N^\alpha}{\beta uh(h-1)}\right) \log^2 N$. By Lemma A.1, it holds that

$$\begin{aligned} \mathbb{P}\{V_1(\mathbf{S}) > B_1 + 2v_{\max}j_1\} &\leq \left(\frac{f_{\max}}{f_{\max} + \gamma}\right)^{j_1} + \left(\frac{\delta}{\gamma} + 1\right) \mathbb{P}\{\mathbf{S} \notin \mathcal{E}_1\} \\ &\leq \left(\frac{f_{\max}}{f_{\max} + \gamma}\right)^{j_1} + \frac{h}{h-1} p_2. \end{aligned}$$

Note that when N is large enough, $\left(\frac{f_{\max}}{f_{\max} + \gamma}\right)^{j_1} \leq (1 + uh\beta N^{-\alpha}(h-1))^{-\left(1 + N^\alpha \frac{1}{\beta uh(h-1)}\right) \log^2 N} \leq e^{-\log^2 N}$. As a result,

$$\mathbb{P}\{V_1(\mathbf{S}) > B_1 + 2v_{\max}j_1\} \leq N^{-\log N} + \frac{h}{h-1} p_2.$$

Since $0 < \alpha < 0.5$ and $k = e^{o(\sqrt{\log N})}$, we have $B_1 + 2v_{\max}j_1 = 1 - h\beta N^{-\alpha} + 2\frac{k}{N} \left(1 + \frac{N^\alpha}{\beta uh(h-1)}\right) \log^2 N < 1 - (h-1)\beta N^{-\alpha}$ when N is large enough. It then follows that

$$\mathbb{P}\{V_1(\mathbf{S}) > 1 - (h-1)\beta N^{-\alpha}\} \leq \mathbb{P}\{V_1(\mathbf{S}) > B_1 + 2v_{\max}j_1\} \leq N^{-\log N} + \frac{h}{h-1} p_2.$$

We now combine the bound above with the following bound given by Lemma B.1:

$$\mathbb{P}\{V_1(\mathbf{S}) > 1 - (h-1)\beta N^{-\alpha}\} \geq 1 - \frac{1}{h-1}.$$

Therefore, $\frac{h}{h-1}p_2 + N^{-\log N} \geq \frac{h-2}{h-1}$, and thus

$$\mathbb{P}\{S_1 - S_2 \leq uh^2\beta N^{-\alpha}\} = p_2 \geq \frac{h-2}{h} - N^{-\log N}.$$

Let $b_q = u^{q-1}h^q\beta N^{-\alpha}$ for an integer $q > 0$. Define a sequence a_q , such that $a_1 = 0, a_2 = 1$ and $a_q = (q-2)a_{q-1} + 2$ for $q > 2$. With this notation, the lower bound above on p_2 can be rewritten as $\mathbb{P}\{S_1 - S_2 \leq a_2 b_2\} \geq \frac{h-2}{h} - N^{-\log N}$. We can use Lemma B.3 inductively to show that for all q with $2 \leq q \leq h$,

$$\mathbb{P}\{S_1 - S_q \leq a_q b_q\} \geq \left(\frac{h-2}{h}\right)^{q-1} - (q-1)N^{-\log N}.$$

Let us condition on $S_1 - S_h \leq a_h b_h$. For ease of notation, let $p_c = \left(\frac{h-2}{h}\right)^{h-1} - (h-1)N^{-\log N}$, which is a lower bound on the probability of the condition. Note that

$$\mathbb{E}[S_1] \leq \mathbb{E}[S_1 \mid S_1 - S_h \leq a_h b_h] \cdot \mathbb{P}\{S_1 - S_h \leq a_h b_h\} + 1 \cdot \mathbb{P}\{S_1 - S_h > a_h b_h\}.$$

Thus

$$\begin{aligned} \mathbb{E}[S_1 \mid S_1 - S_h \leq a_h b_h] &\geq \frac{1 - \beta N^{-\alpha} - (1 - \mathbb{P}\{S_1 - S_h \leq a_h b_h\})}{\mathbb{P}\{S_1 - S_h \leq a_h b_h\}} \\ &\geq 1 - \frac{\beta}{p_c} N^{-\alpha}. \end{aligned}$$

We can also see that

$$\begin{aligned} &\mathbb{P}\left\{\sum_{i=1}^h S_i \geq h - \frac{1}{3d}\right\} \\ &\geq \mathbb{P}\left\{\sum_{i=1}^h S_i \geq h - \frac{1}{3d} \mid S_1 - S_h \leq a_h b_h\right\} \mathbb{P}\{S_1 - S_h \leq a_h b_h\} \\ &\geq p_c \mathbb{P}\left\{hS_1 - h(S_1 - S_h) \geq h - \frac{1}{3d} \mid S_1 - S_h \leq a_h b_h\right\} \\ &\geq p_c \mathbb{P}\left\{S_1 \geq 1 - \frac{1}{3dh} + a_h b_h \mid S_1 - S_h \leq a_h b_h\right\}. \end{aligned} \tag{24}$$

Utilizing the Markov inequality gives

$$\begin{aligned} (24) &\geq p_c \left(1 - \frac{3dh - 3dh\mathbb{E}[S_1 \mid S_1 - S_h \leq a_h b_h]}{1 - 3dha_h b_h}\right) \\ &\geq p_c \left(1 - \frac{\beta}{p_c} \frac{3dh}{1 - 3dha_h b_h} N^{-\alpha}\right). \end{aligned}$$

Recall that $a_q = (q-2)a_{q-1} + 2$ for $q > 2$ and $a_2 = 1$. We have $a_h \leq 2h^h$, and thus $a_h b_h \leq 2\beta u^h h^{2h} N^{-\alpha}$. As $d = e^{o(\log N / \log k)}$, $k = e^{o(\sqrt{\log N})}$, $h = O(\log k)$, we have $\ln(a_h b_h) = -\Omega(\log N)$. Furthermore, since $\ln(3dh) = o(\log N / \log k) + O(\log k)$, $\alpha > 0$, it holds

$$1 - \frac{\beta}{p_c} \frac{3dh}{1 - 3dha_h b_h} N^{-\alpha} \geq \frac{1}{2}$$

if N is sufficiently large. Note that p_c is equal to $\left(\frac{h-2}{h}\right)^{h-1} - (h-1)N^{-\log N}$ which converges to $\frac{1}{e^2}$. We could conclude that when N goes to infinity, we have

$$\mathbb{P}\left\{\sum_{i=1}^h S_i \geq h - \frac{1}{3d}\right\} \geq \frac{1}{4e^2}.$$

□

5.2 Proof Of Theorem 3.4

PROOF. Let $h = 12e^2 \ln k$. Then $h = O(\log k)$. Suppose that we have an incoming job. By Theorem 3.3 and the PASTA property of a Poisson arrival process, with probability at least $\frac{1}{4e^2}$, this job will see a state s such that $\sum_{i=1}^h s_i \geq h - \frac{1}{3d}$. By Lemma B.4, the dispatcher will route at least one task of this job into a queue of length at least $h + 1$ with probability $1 - o(1)$. Let T be the delay of the job. Then it holds for a large enough N ,

$$\mathbb{E}[T] \geq 3 \ln k(1 - o(1)) \geq 2 \ln k,$$

which completes the proof. □

6 DISCUSSION ON AN ALTERNATIVE NOTION OF ZERO QUEUEING DELAY

In this section, we consider an alternative notion of zero queueing delay that may be of interest and may provide more understanding into the dynamics of systems with parallel jobs. We will refer to this alternative notion as *zero waiting* to differentiate it from the zero queueing delay we consider in the main part of the paper. We say that zero waiting is achieved if in steady state, all the tasks of an incoming job enter service immediately upon arrival without waiting in queues with high probability as $N \rightarrow \infty$. It is easy to see that zero waiting is a much stronger requirement than zero queueing delay. Indeed, we show in Theorem 6.1 below, the minimum probe overhead needed for achieving zero waiting is larger than $\frac{1}{2(1-\lambda)}$, which is in the same order as the value in the impossibility results for non-parallel jobs. The proof of Theorem 6.1 is straightforward and given in Appendix C.

Note that although this notion of zero waiting for parallel jobs seems to resemble the zero queueing delay for non-parallel jobs, the two systems have fundamentally different dynamics and thus it is hard to directly compare these two notion. For parallel jobs, a batch of tasks arrive together and zero waiting requires all of them to be assigned to idle servers *simultaneously*. In contrast, for non-parallel jobs, there is no concept of batches. The single-task jobs arrive one by one and zero queueing delay requires a job to be assigned to an idle server when it arrives.

THEOREM 6.1. *Consider a system with N servers where each job consists of k tasks. Let the load be $\lambda = 1 - \beta N^{-\alpha}$ with $0 < \beta \leq 1$ and $\alpha \geq 0$. Assume that the buffers have unlimited sizes. Under the batch-filling policy with a probe overhead d such that $1 \leq d \leq \frac{1}{2(1-\lambda)}$, the probability in steady state that all the tasks of an incoming job are assigned to idle servers is smaller than or equal to 0.5.*

7 SIMULATION RESULTS

In this section, we perform two sets of simulations to demonstrate our theoretical results and explore settings beyond those in our theoretical analysis. The first set illustrates the scaling behavior of the system as N grows under various probe ratios, and investigates the gap between our achievability results and impossibility results. The second set of simulations experiment on more general service time distributions beyond the exponential distribution and correlation among task service times.

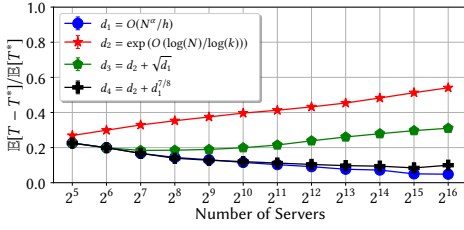


Fig. 3. Queueing delays under different probe ratios: d_1 is sufficient for convergence to zero queueing delay; $d_1 > d_4 > d_3 > d_2$.

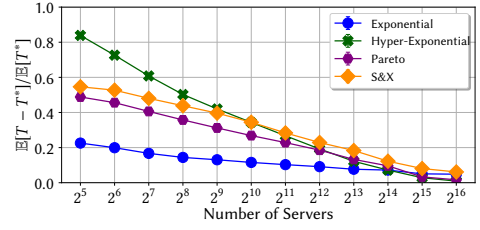


Fig. 4. Queueing delays under different service time distributions.

7.1 Scaling Behavior with Various Probe Ratios

This set of simulations use the setting of our theoretical results with $\lambda = 1 - N^{-0.3}$ ($\alpha = 0.3$ and $\beta = 1$). We let k , the number of tasks per job, scale with N as $k = \lfloor \ln^2 N \rfloor$. The values for N and the corresponding k used in the simulations are given in Table 1. These values are reasonable in practice considering that datacenters nowadays typically have tens of thousands of nodes (each with multiple cores) per cluster and a job may consist of hundreds of tasks [1].

N	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536
k	12	17	23	30	38	48	58	69	81	94	108	122

Table 1. Scaling parameters

We explore four scaling settings of the probe ratio. The first setting uses a probe ratio of $d_1 = \frac{4}{(1-\lambda)h} = \frac{4N^\alpha}{h}$ with $h = \lceil \frac{\log k}{\log \log k} \rceil$, which satisfies the conditions in Theorems 3.1 and 3.2 to achieve zero queueing delay. The second setting uses a probe ratio of $d_2 = \exp(0.5 \log N / \log k)$, which is slightly larger than the value in Theorems 3.3 and 3.4 that guarantees non-zero queueing delay. The other two settings use probe ratio values d_3 and d_4 that interpolate between d_1 and d_2 to investigate the threshold under which the system transits from zero queueing delay to non-zero queueing delay. Note that $d_1 > d_4 > d_3 > d_2$ for all values of N in the simulations. More details of the settings can be found in Appendix D.1.

Figure 3 shows the simulation results for the queueing delay $\frac{\mathbb{E}[T - T^*]}{\mathbb{E}[T^]}$, where the results are averaged over ten independent runs. Please refer to Appendix D.2 for the exact values and standard deviations. Since the standard deviations are very small ($\sim 10^{-4}$), the error bars are not visible in the plots. The curve for d_1 demonstrates the trend for the queueing delay to converge to zero as predicted by the theoretical results. It does not exactly reach zero but becomes reasonably close. Under the probe ratios d_2 and d_3 , the queueing delay clearly deviates from zero. Under the probe ratio d_4 , the queueing delay flattens out after some initial drop as N becomes large. Therefore, it is plausible that the transition from zero queueing delay to non-zero queueing delay happens at a probe ratio value near d_4 . Since d_4 is much closer to d_1 than to d_2 , we expect our impossibility results to be not tight. Pinning down the exact threshold for the transition (or proving the nonexistence of such a threshold) is of great theoretical interest and we leave it for future research.

To further investigate how k , the number of tasks per job, affects the scaling behavior under different probe ratios, we examine another setting where $k = \lfloor \sqrt{N} \rfloor$. This scaling of k is beyond our theoretical framework, but the queueing delays exhibit similar trends as those in the setting where $k = \lfloor \ln^2 N \rfloor$. Details of the simulation results are given in Appendix D.3.

7.2 More General Settings for Task Service Times

This set of simulations explore distributions beyond the exponential distribution for task service times and correlation among their service times. Figure 4 shows the results for four settings: (1) *i.i.d. exponential distribution* with rate 1 (denoted as Exp(1)). This is the baseline distribution that is assumed for our theoretical analysis. (2) *i.i.d. bounded Pareto* in range $[1, 1000]$ with a shape constant 1.5. (3) *i.i.d. hyper-exponential* that follows Exp(1) with probability 0.99 and Exp(0.01) with probability 0.01. We re-scale the arrival rates so all the systems have the same load $\lambda = 1 - N^{-0.3}$. (4) *S&X model* for correlated task service times, which is a model proposed in [13] and has been extensively studied since then. In the S&X model, the service time of the each task in a job can be written as $S \cdot X$, where every task in the same job shares the same X , but different tasks have their own S 's that are independent among tasks. Here we assume S and X are both exponentially distributed with rate 1. The probe overhead is chosen to be the d_1 in Section 7.1 such that zero queueing delay is provably achievable under the exponential distribution.

We observe that empirically, the queueing delay has a trend that approaches zero under all the four settings, despite of the larger coefficients of variation for the bounded Pareto and hyper-exponential distributions and the correlation among task service times in the S&X model. These simulation results suggest that our theoretical results have some robustness with respect to service time distributions and correlations. We comment that there is little existing work on zero queueing delay for general service time distributions with the exception of [24], which studies the Coxian-2 distribution for non-parallel jobs. Generalizing our analysis to general service time distributions with possible correlations is a research direction that deserves much further effort, as it is for many problems in queueing systems.

8 CONCLUSIONS

We studied queueing delay in a system where jobs consist of parallel tasks. We first proposed a notion of zero queueing delay in a relative sense for such parallel jobs. We then derived conditions on the probe overhead for achieving zero queueing delay and for guaranteeing non-zero queueing delay. One interesting implication of the results is that under parallelization, the probe overhead needed for achieving zero queueing delay is lower than that in a system with non-parallel (single-task) jobs under the same load. Through simulations, we demonstrated that the numerical results are consistent with the theoretical results under reasonable settings, and investigated several questions that are hard to answer analytically.

Acknowledgment: The work of Wentao Weng was conducted during a visit to the Computer Science Department, CMU in 2019.

REFERENCES

- [1] George Amvrosiadis, Jun Woo Park, Gregory R Ganger, Garth A Gibson, Elisabeth Baseman, and Nathan DeBardeleben. 2018. On the diversity of cluster workloads and its impact on research results. In *Proc. USENIX Ann. Technical Conf. (ATC)*. 533–546.
- [2] Sayan Banerjee and Debankur Mukherjee. 2019. Join-the-shortest queue diffusion limit in Halfin–Whitt regime: Tail asymptotics and scaling of extrema. *Ann. Appl. Probab.* 29, 2 (2019), 1262–1309.
- [3] 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.
- [4] Eric Boutin, Jaliya Ekanayake, Wei Lin, Bing Shi, Jingren Zhou, Zhengping Qian, Ming Wu, and Lidong Zhou. 2014. Apollo: Scalable and coordinated scheduling for cloud-scale computing. In *Proc. USENIX Conf. Operating Systems Design and Implementation (OSDI)*. USENIX, 285–300.
- [5] Anton Braverman. 2018. Steady-state analysis of the Join the Shortest Queue model in the Halfin-Whitt regime. *arXiv:1801.05121 [math.PR]* (2018).

- [6] Anton Braverman and JG Dai. 2017. Stein’s method for steady-state diffusion approximations of $M/Ph/n + M$ systems. *Ann. Appl. Probab.* 27 (Feb. 2017), 550–581. <https://doi.org/10.1214/16-AAP1211>
- [7] Anton Braverman, JG 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.
- [8] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. 2007. Dynamo: amazon’s highly available key-value store. *SIGOPS Oper. Syst. Rev.* 41, 6 (2007), 205–220.
- [9] Christina Delimitrou, Daniel Sanchez, and Christos Kozyrakis. 2015. Tarcil: reconciling scheduling speed and quality in large shared clusters. In *Proc. ACM Symp. Cloud Computing (SOCC)*. 97–110.
- [10] Patrick Eschenfeldt and David Gamarnik. 2018. Join the shortest queue with many servers. The heavy-traffic asymptotics. *Math. Oper. Res.* 43, 3 (2018), 867–886.
- [11] Stewart N. Ethier and Thomas G. Kurtz. 1986. *Markov Processes: Characterization and Convergence*. John Wiley & Sons, New York.
- [12] David Gamarnik, John N Tsitsiklis, and Martin Zubeldia. 2016. Delay, memory, and messaging tradeoffs in distributed service systems. In *Proc. ACM SIGMETRICS/PERFORMANCE Jt. Int. Conf. Measurement and Modeling of Computer Systems*. ACM, 1–12.
- [13] Kristen Gardner, Mor Harchol-Balter, and Alan Scheller-Wolf. 2016. A Better Model for Job Redundancy: Decoupling Server Slowdown and Job Size. In *IEEE Int. Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. London, United Kingdom, 1–10.
- [14] Nicolas Gast. 2017. Expected Values Estimated via Mean-Field Approximation are $1/N$ -Accurate. In *Proc. ACM Measurement and Analysis of Computing Systems (POMACS)*, Vol. 45. ACM, 50–50.
- [15] Nicolas Gast and Benny Van Houdt. 2017. A refined mean field approximation. In *Proc. ACM Measurement and Analysis of Computing Systems (POMACS)*, Vol. 1. ACM, 33.
- [16] Ionel Gog, Malte Schwarzkopf, Adam Gleave, Robert NM Watson, and Steven Hand. 2016. Firmament: Fast, centralized cluster scheduling at scale. In *Proc. USENIX Conf. Operating Systems Design and Implementation (OSDI)*. USENIX, 99–115.
- [17] Varun Gupta and Neil Walton. 2019. Load Balancing in the Nondegenerate Slowdown Regime. *Oper. Res.* 67, 1 (2019), 281–294.
- [18] Itai Gurvich. 2014. Diffusion models and steady-state approximations for exponentially ergodic Markovian queues. *Ann. Appl. Probab.* 24, 6 (2014), 2527–2559.
- [19] Shlomo Halfin and Ward Whitt. 1981. Heavy-traffic limits for queues with many exponential servers. *Oper. Res.* 29, 3 (1981), 567–588.
- [20] Wassily Hoeffding. 1963. Probability Inequalities for Sums of Bounded Random Variables. *J. Amer. Stat. Assoc.* 58, 301 (1963), 13–30. <http://www.jstor.org/stable/2282952>
- [21] Eric Jonas, Qifan Pu, Shivaram Venkataraman, Ion Stoica, and Benjamin Recht. 2017. Occupy the cloud: Distributed computing for the 99%. In *Proc. ACM Symp. Cloud Computing (SOCC)*. 445–451.
- [22] Avinash Lakshman and Prashant Malik. 2010. Cassandra: a decentralized structured storage system. *SIGOPS Oper. Syst. Rev.* 44, 2 (2010), 35–40.
- [23] Xin Liu and Lei Ying. 2018. On achieving zero delay with power-of-d-choices load balancing. In *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*. Honolulu, HI, USA, 297–305.
- [24] Xin Liu and Lei Ying. 2019. On Universal Scaling of Distributed Queues under Load Balancing. *arXiv:1912.11904 [math.PR]* (2019).
- [25] 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.
- [26] 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.
- [27] Michael Mitzenmacher. 2001. The power of two choices in randomized load balancing. *IEEE Trans. Parallel Distrib. Syst.* 12, 10 (2001), 1094–1104.
- [28] Debankur Mukherjee, Sem C Borst, Johan SH Van Leeuwen, and Philip A Whiting. 2018. Universality of power-of-d load balancing in many-server systems. *Stoch. Syst.* 8, 4 (2018), 265–292.
- [29] Willie Neiswanger, Chong Wang, and Eric Xing. 2013. Asymptotically exact, embarrassingly parallel MCMC. *arXiv:1311.4780 [stat.ML]* (2013).
- [30] Kay Ousterhout, Aurojit Panda, Joshua Rosen, Shivaram Venkataraman, Reynold Xin, Sylvia Ratnasamy, Scott Shenker, and Ion Stoica. 2013. The case for tiny tasks in compute clusters. In *Proc. USENIX Conf. Hot Topics in Operating Systems (HotOS)*.
- [31] Kay Ousterhout, Patrick Wendell, Matei Zaharia, and Ion Stoica. 2013. Sparrow: distributed, low latency scheduling. In *Proc. ACM Symp. Operating Systems Principles (SOSP)*. ACM, 69–84.

- [32] Seva Shneer and Alexander Stolyar. 2020. Large-scale parallel server system with multi-component jobs. *arXiv:2006.11256 [math.PR]* (2020).
- [33] Charles Stein. 1972. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proc. 6th Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*. The Regents of the University of California.
- [34] Alexander L Stolyar. 2015. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Syst.* 80, 4 (2015), 341–361.
- [35] Alexander L. Stolyar. 2015. Tightness of Stationary Distributions of a Flexible-Server System in the Halfin-Whitt Asymptotic Regime. *Stoch. Syst.* 5, 2 (2015), 239–267.
- [36] Vinod Kumar Vavilapalli, Arun C. Murthy, Chris Douglas, Sharad Agarwal, Mahadev Konar, Robert Evans, Thomas Graves, Jason Lowe, Hitesh Shah, Siddharth Seth, Bikas Saha, Carlo Curino, Owen O’Malley, Sanjay Radia, Benjamin Reed, and Eric Baldeschwieler. 2013. Apache Hadoop YARN: Yet Another Resource Negotiator. In *Proc. ACM Symp. Cloud Computing (SOCC)* (Santa Clara, California). ACM, New York, NY, USA.
- [37] Abhishek Verma, Luis Pedrosa, Madhukar R. 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).
- [38] Nikita Dmitrievna Vvedenskaya, Roland L’vovich Dobrushin, and Fridrikh Izrailevich Karpelevich. 1996. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problems of Information Transmission* 32, 1 (1996), 15–27.
- [39] Weina Wang, Mor Harchol-Balter, Haotian Jiang, Alan Scheller-Wolf, and R. Srikant. 2019. Delay asymptotics and bounds for multitask parallel jobs. *Queueing Syst.* 91, 3 (01 April 2019), 207–239.
- [40] 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. In *Proc. ACM SIGMETRICS Int. Conf. Measurement and Modeling of Computer Systems*, Vol. 45. ACM, 232–245.
- [41] Richard R Weber. 1978. On the optimal assignment of customers to parallel servers. *J. Appl. Probab.* 15, 2 (1978), 406–413.
- [42] Wayne Winston. 1977. Optimality of the shortest line discipline. *J. Appl. Probab.* 14, 1 (1977), 181–189.
- [43] Lei Ying. 2016. On the approximation error of mean-field models. *ACM SIGMETRICS Perform. Evaluation Rev.* 44, 1 (2016), 285–297.
- [44] Lei Ying. 2017. Stein’s method for mean field approximations in light and heavy traffic regimes. *ACM SIGMETRICS Perform. Evaluation Rev.* 45, 1 (2017), 49.
- [45] Lei Ying, R. Srikant, and Xiaohan Kang. 2015. The power of slightly more than one sample in randomized load balancing. In *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*. Kowloon, Hong Kong, 1131–1139.
- [46] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: Cluster Computing with Working Sets. In *Proc. USENIX Conf. Hot Topics in Cloud Computing (HotCloud)* (Boston, MA). USENIX, USA.

A PROOFS OF LEMMAS 4.1, 4.2 AND 4.3

A.1 Proof of Lemma 4.1

LEMMA 4.1 [RESTATED]. Consider m independent random variables Y_1, \dots, Y_m with $m \leq k$, where each Y_i ($1 \leq i \leq m$) is the sum of n_i i.i.d. random variables that follow the exponential distribution with rate 1. In the asymptotic regime that k goes to infinity, if $\max \{n_1, \dots, n_m\} = o(\log k)$, then

$$\mathbb{E}[\max \{Y_1, \dots, Y_m\}] \leq \ln k + o(\ln k).$$

PROOF. The general proof idea is folklore, but here we derive the exact bounds for our purpose. Let $M_X(s)$ be the moment generating function of a random variable X . By assumption, $Y_i = \sum_{j=1}^{n_i} X_{i,j}$, and $X_{i,j}$, $1 \leq i \leq m$, $1 \leq j \leq n_i$ are all independent and exponentially distributed with mean 1. Therefore, for any $1 \leq i \leq m$, $1 \leq j \leq n_i$ and any s with $0 < s < 1$,

$$M_{X_{i,j}}(s) = \mathbb{E} \left[e^{sX_{i,j}} \right] = \frac{1}{1-s},$$

$$M_{Y_i}(s) = \mathbb{E} \left[e^{sY_i} \right] = \left(\frac{1}{1-s} \right)^{n_i}.$$

Let $q = \max \{n_1, \dots, n_m\}$. It holds that for any s with $0 < s < 1$,

$$\exp \left(s \mathbb{E} \left[\max_{j=1}^m Y_j \right] \right) \leq \mathbb{E} \left[\exp(s \max_{j=1}^m Y_j) \right] \quad (25)$$

$$= \mathbb{E} \left[\max_{j=1}^m \exp(s Y_j) \right] \quad (26)$$

$$\leq \sum_{j=1}^m \mathbb{E} [\exp(s Y_j)] \quad (27)$$

$$\leq m \left(\frac{1}{1-s} \right)^q, \quad (28)$$

where (25) is due to Jensen's inequality and (27) is true since the maximum is upper bounded by the sum. As a result,

$$\mathbb{E} \left[\max_{j=1}^m Y_j \right] \leq \frac{\ln m}{s} + q \cdot \frac{-\ln(1-s)}{s} \quad (29)$$

$$\leq \frac{\ln k}{s} + q \cdot \frac{-\ln(1-s)}{s}, \quad (30)$$

where we have used the assumption that $m \leq k$. Since we assume that $q = o(\log k)$, we can write q as $q = \ell(k) \ln k$ where $\ell(k) \rightarrow 0^+$ as $k \rightarrow \infty$. Let $s = 1 - \ell(k)$ in (30), then

$$\mathbb{E} \left[\max_{j=1}^m Y_j \right] \leq \frac{\ln k}{1 - \ell(k)} (1 - \ell(k) \ln \ell(k)) \quad (31)$$

$$= (\ln k) \left(1 + \frac{\ell(k)}{1 - \ell(k)} \right) (1 - \ell(k) \ln \ell(k)). \quad (32)$$

Note that $\lim_{k \rightarrow \infty} \ell(k) \ln \ell(k) = 0$. Then as $k \rightarrow \infty$,

$$\mathbb{E} \left[\max_{j=1}^m Y_j \right] \leq (\ln k)(1 + o(1)),$$

which completes the proof. \square

A.2 Proof of Lemma 4.2 (Filling Probability)

LEMMA 4.2 (FILLING PROBABILITY) [RESTATE]. *Under the assumptions of Theorem 3.1, given that the system is in a state \mathbf{s} such that*

$$\sum_{i=1}^{\ell} s_i \leq \ell \left(1 - \frac{1}{4} \beta N^{-\alpha} \right), \quad (33)$$

the probability of the event FILL_{ℓ} for any $\ell \in \{h-1, h, b\}$ can be bounded as $\mathbb{P} \{ \text{FILL}_{\ell} \} \geq 1 - \frac{1}{N}$ when N is sufficiently large.

PROOF. Assume that a job arrival sees a state $\mathbf{S} = \mathbf{s}$ that satisfies

$$\sum_{i=1}^{\ell} s_i \leq \ell \left(1 - \frac{1}{4} \beta N^{-\alpha} \right).$$

We focus on the the number of spaces below the threshold ℓ in the sampled queues, denoted by N_{ℓ} . Then N_{ℓ} is the maximum number of tasks that can be put into these queues such that all of these tasks are at queuing positions below ℓ . Therefore,

$$\mathbb{P} \{ \text{FILL}_{\ell} \} = \mathbb{P} \{ N_{\ell} \geq k \} \geq 1 - \mathbb{P} \{ N_{\ell} \leq k \}.$$

Now we bound $\mathbb{P}\{N_\ell \leq k\}$. We can think of the sampling process of batch-filling as sampling kd queues one by one without replacement. Let X_1, X_2, \dots, X_{kd} be the numbers of spaces below ℓ in the 1st, 2nd, \dots , kd th sampled queues, respectively. Then $N_\ell = X_1 + \dots + X_{kd}$. It is not hard to see that for each of the sampled queue and each integer x with $1 \leq x \leq \ell$,

$$\mathbb{P}\{X_i = x\} = s_{\ell-x} - s_{\ell-x+1},$$

and $\mathbb{P}\{X_i = 0\} = s_\ell$.

Note that since we sample without replacement, X_1, X_2, \dots, X_{kd} are not independent. But we can still derive concentration bounds using a result of Hoeffding [20, Theorem 4]. By this result, we have $\mathbb{E}\left[f\left(\sum_{i=1}^{kd} X_i\right)\right] \leq \mathbb{E}\left[f\left(\sum_{i=1}^{kd} Y_i\right)\right]$ for any continuous and convex function $f(\cdot)$, where Y_1, Y_2, \dots, Y_{kd} are i.i.d. and follow the same distribution as X_1 . We take the function $f(\cdot)$ to be $f(x) = e^{-tx}$ with $t > 0$. Then

$$\begin{aligned} & \mathbb{P}\{N_\ell \leq k\} \\ &= \mathbb{P}\left\{e^{-tN_\ell} \geq e^{-tk}\right\} \\ &\leq e^{tk} \prod_{i=1}^{kd} \mathbb{E}\left[e^{-tY_i}\right] \\ &= e^{tk} \prod_{i=1}^{kd} \left(1 - \sum_{j=1}^{\ell} (s_{\ell-j} - s_{\ell-j+1}) (1 - e^{-tj})\right). \end{aligned}$$

Since $1 - x \leq e^{-x}$ for each $x \geq 0$, this can be further bounded as

$$\begin{aligned} & \mathbb{P}\{N_\ell \leq k\} \\ &\leq \exp\left(tk - kd \sum_{j=1}^{\ell} (s_{\ell-j} - s_{\ell-j+1}) (1 - e^{-tj})\right) \\ &\leq \exp\left(tk + kd \sum_{j=1}^{\ell} (s_{j-1} - s_j) \left(e^{-t(\ell-j+1)} - 1\right)\right). \end{aligned} \quad (34)$$

Rearranging the terms in the sum in (34), we get

$$\begin{aligned} & \sum_{j=1}^{\ell} (s_{j-1} - s_j) \left(e^{-t(\ell-j+1)} - 1\right) \\ &= \left(e^{-t\ell} - 1\right) + (e^t - 1) \sum_{j=1}^{\ell} s_j e^{-t(\ell-j+1)}. \end{aligned} \quad (35)$$

Since $1 \geq s_1 \geq \dots \geq s_\ell$ and we have assumed that $\sum_{j=1}^{\ell} s_j \leq \ell \left(1 - \frac{1}{4}\beta N^{-\alpha}\right)$, (35) is maximized when

$$s_1 = s_2 = \dots = s_\ell = 1 - \frac{1}{4}\beta N^{-\alpha}.$$

Therefore, the upper bound becomes

$$\mathbb{P}\{N_\ell \leq k\} \leq \exp\left(tk + kd \left(e^{-t\ell} - 1\right) \frac{1}{4}\beta N^{-\alpha}\right).$$

Now we apply the condition that $d \geq \frac{8N^\alpha}{\beta h}$ and let $t = \frac{\ln(2\ell) - \ln h}{\ell}$. Then

$$\mathbb{P}\{N_\ell \leq k\}$$

$$\begin{aligned} &\leq \exp\left(tk + \frac{2k}{h} \left(e^{-t\ell} - 1\right)\right) \\ &= \exp\left(\frac{k}{h} \left(\frac{h}{\ell} (\ln(2\ell) - \ln h) + \frac{h}{\ell} - 2\right)\right). \end{aligned}$$

Recall the we have assumed that $\frac{k}{h} = \omega(\log N)$ and $h = \omega(1)$. Then it can be verified that with a sufficiently large N , $\frac{h}{\ell} (\ln(2\ell) - \ln h) + \frac{h}{\ell} + 2N^{-0.5} - 2$ is smaller than a negative constant for all $\ell \in \{h-1, h, b\}$. Thus

$$\mathbb{P}\{N_\ell \leq k\} \leq \exp(-\omega(\log N)) \leq \frac{1}{N}.$$

As a result,

$$\mathbb{P}\{\text{FILL}_\ell\} \geq 1 - \mathbb{P}\{N_\ell \leq k\} \geq 1 - \frac{1}{N},$$

which completes the proof. \square

A.3 Proof of Lemma 4.3

Our proof of Lemma 4.3 relies on Lemma A.1 below. Lemma A.1 slightly generalizes the well-known Lyapunov-based tail bounds (see, e.g., [40], [24] and [3]) in that it allows different drift bounds depending on whether a state \mathbf{s} is in a set \mathcal{E} or not. In our proof of Lemma 4.3, we only need to let \mathcal{E} be the whole state space. But this generalization will be needed in the proof of impossibility results in Section 5. We omit the proof of Lemma A.1 since it only needs minor modification to the arguments used in proving the well-known existing bounds.

LEMMA A.1. *Consider a continuous time Markov chain $\{\mathbf{S}(t) : t \geq 0\}$ with a finite state space \mathcal{S} and a unique stationary distribution π . For a Lyapunov function $V : \mathcal{S} \rightarrow [0, +\infty)$, define the drift of V at a state $\mathbf{s} \in \mathcal{S}$ as*

$$\Delta V(\mathbf{s}) = \sum_{\mathbf{s}' \in \mathcal{S}, \mathbf{s} \neq \mathbf{s}'} r_{\mathbf{s} \rightarrow \mathbf{s}'} (V(\mathbf{s}') - V(\mathbf{s})),$$

where $r_{\mathbf{s} \rightarrow \mathbf{s}'}$ is the transition rate from state \mathbf{s} to \mathbf{s}' . Suppose that

$$\begin{aligned} v_{\max} &:= \sup_{\mathbf{s}, \mathbf{s}' \in \mathcal{S}: r_{\mathbf{s} \rightarrow \mathbf{s}'} > 0} |V(\mathbf{s}) - V(\mathbf{s}')| < \infty \\ f_{\max} &:= \max \left\{ 0, \sup_{\mathbf{s} \in \mathcal{S}} \sum_{\mathbf{s}': V(\mathbf{s}') > V(\mathbf{s})} r_{\mathbf{s} \rightarrow \mathbf{s}'} (V(\mathbf{s}') - V(\mathbf{s})) \right\} < \infty. \end{aligned}$$

Then if there is a set \mathcal{E} with $B > 0, \gamma > 0, \delta \geq 0$ such that

- $\Delta V(\mathbf{s}) \leq -\gamma$ when $V(\mathbf{s}) \geq B$ and $\mathbf{s} \in \mathcal{E}$,
- $\Delta V(\mathbf{s}) \leq \delta$ when $V(\mathbf{s}) \geq B$ and $\mathbf{s} \notin \mathcal{E}$,

it holds that for all $j \in \mathbb{N}$,

$$\mathbb{P}\{V(\mathbf{s}) \geq B + 2v_{\max}j\} \leq \left(\frac{f_{\max}}{f_{\max} + \gamma}\right)^j + \left(\frac{\delta}{\gamma} + 1\right) \mathbb{P}\{\mathbf{s} \notin \mathcal{E}\}.$$

Now we are ready to present to proof of Lemma 4.3.

LEMMA 4.3 (STATE-SPACE COLLAPSE) [RESTATED]. *Under the assumption of Theorem 3.1, consider the following Lyapunov function:*

$$V(\mathbf{s}) = \min \left\{ \frac{1}{h-1} \sum_{i=h}^b s_i, b \left(\left(1 - \frac{1}{2} \beta N^{-\alpha}\right) - \frac{1}{h-1} \sum_{i=1}^{h-1} s_i \right)^+ \right\},$$

where the superscript $+$ denotes the function $x^+ = \max\{x, 0\}$. Let $B = \frac{b-h+1}{h-1} \left(\beta N^{-\alpha} + \frac{\log N}{\sqrt{N}} \right)$. Then for any state \mathbf{s} such that $V(\mathbf{s}) > B$, its Lyapunov drift can be upper bounded as follows

$$\Delta V(\mathbf{s}) = GV(\mathbf{s}) \leq -\frac{b}{\sqrt{N}}.$$

Consequently, when N is sufficiently large,

$$\mathbb{P} \left\{ V(\mathbf{S}) > B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}} \right\} \leq e^{-\frac{1}{2} \log^2 N}.$$

PROOF. Consider the Lyapunov function in the lemma, i.e.,

$$V(\mathbf{s}) = \min \left\{ \frac{1}{h-1} \sum_{i=h}^b s_i, b \left(\left(1 - \frac{1}{2} \beta N^{-\alpha} \right) - \frac{1}{h-1} \sum_{i=1}^{h-1} s_i \right)^+ \right\}.$$

We will refer to the first term and second term in the minimum as \mathcal{T}_1 and \mathcal{T}_2 , respectively. Let $B = \frac{b-h+1}{h-1} \left(\beta N^{-\alpha} + \frac{\log N}{\sqrt{N}} \right)$ and suppose $V(\mathbf{s}) > B$. Recall that the drift of V is given by

$$\Delta V(\mathbf{s}) = GV(\mathbf{s}) = \sum_{\mathbf{s}' \in \mathcal{S}, \mathbf{s} \neq \mathbf{s}'} r_{\mathbf{s} \rightarrow \mathbf{s}'} (V(\mathbf{s}') - V(\mathbf{s})),$$

where $r_{\mathbf{s} \rightarrow \mathbf{s}'}$ is the transition rate from state \mathbf{s} to \mathbf{s}' . Let $\mathbf{e}_i = (0, \dots, 0, \frac{1}{N}, 0, \dots, 0)$ be a vector of length b whose i th entry is $\frac{1}{N}$ and all the other entries are zero. We divide the discussion into two cases.

Case 1: $\mathcal{T}_1 \leq \mathcal{T}_2$. In this case $V(\mathbf{s}) = \mathcal{T}_1$. When the state transition is due to a task departure from a queue of length i , which has a rate of $N(s_i - s_{i+1})$, then

$$V(\mathbf{s} - \mathbf{e}_i) = \begin{cases} V(\mathbf{s}), & \text{if } 1 \leq i < h, \\ V(\mathbf{s}) - \frac{1}{N(h-1)}, & \text{if } h \leq i \leq b. \end{cases}$$

Now consider the state transition due to a job arrival. Let a_i be the queueing position that task i is assigned to. Then the next state can be written as

$$\mathbf{s} + \mathbf{e}_{a_1} + \dots + \mathbf{e}_{a_k}.$$

Note that when the event FILL_{h-1} happens, the dispatcher puts all k tasks to positions below threshold $h-1$. Then under FILL_{h-1} , s_i does not change for $i \geq h$, which implies that

$$V(\mathbf{s} + \mathbf{e}_{a_1} + \dots + \mathbf{e}_{a_k}) = V(\mathbf{s}).$$

We can show that $\mathbb{P}\{\text{FILL}_{h-1}\} \geq 1 - \frac{1}{N}$ using Lemma 4.2 since $\mathcal{T}_2 \geq \mathcal{T}_1 > B > 0$. Otherwise, i.e., when FILL_{h-1} is not true, it is easy to see that

$$V(\mathbf{s} + \mathbf{e}_{a_1} + \dots + \mathbf{e}_{a_k}) \leq V(\mathbf{s}) + \frac{k}{N(h-1)}.$$

Therefore,

$$\begin{aligned} \Delta V(\mathbf{s}) &\leq \sum_{i=1}^b N(s_i - s_{i+1}) (V(\mathbf{s} - \mathbf{e}_i) - V(\mathbf{s})) + \frac{N\lambda}{k} \frac{1}{N} \frac{k}{N(h-1)} \\ &= \frac{1}{N(h-1)} - \frac{s_h}{h-1} \\ &\leq \frac{1}{N(h-1)} - \frac{1}{h-1} \frac{1}{b-h+1} \sum_{i=h}^b s_i. \end{aligned}$$

By the assumption that $\mathcal{T}_1 > B$, we have

$$\frac{1}{b-h+1} \sum_{i=h}^b s_i \geq \frac{h-1}{b-h+1} B = \beta N^{-\alpha} + \frac{\log N}{\sqrt{N}}.$$

Inserting this back to the upper bound on $\Delta V(\mathbf{s})$ gives

$$\Delta V(\mathbf{s}) \leq -\frac{1}{h-1} \left(-\frac{1}{N} + \beta N^{-\alpha} + \frac{\log N}{\sqrt{N}} \right).$$

Since $\frac{\beta N^{-\alpha}}{h-1} \geq \frac{N^{-\alpha}}{k} \geq \frac{b}{\sqrt{N}}$ and $\frac{\log N}{\sqrt{N}} \geq \frac{1}{N}$ when N is sufficiently large, this upper bound becomes

$$\Delta V(\mathbf{s}) \leq -\frac{b}{\sqrt{N}}.$$

Case 2: $\mathcal{T}_1 > \mathcal{T}_2$. In this case $V(\mathbf{s}) = \mathcal{T}_2$. Similarly, a task departs from a queue of length i at a rate of $N(s_i - s_{i+1})$. The change in $V(\mathbf{s})$ can be bounded as

$$V(\mathbf{s} - e_i) - V(\mathbf{s}) \leq \begin{cases} \frac{b}{N(h-1)}, & \text{if } 1 \leq i < h, \\ 0, & \text{if } h \leq i \leq b. \end{cases}$$

When a job arrives, under the event FILL_{h-1} ,

$$V(\mathbf{s} + e_{a_1} + \cdots + e_{a_k}) = V(\mathbf{s}) - \frac{kb}{N(h-1)},$$

where we have used the fact that $\mathcal{T}_2 > B$. Again, $\mathbb{P}\{\text{FILL}_{h-1}\} \geq 1 - \frac{1}{N}$ by Lemma 4.2. Otherwise, i.e., when FILL_{h-1} is not true, $V(\mathbf{s} + e_{a_1} + \cdots + e_{a_k}) \leq V(\mathbf{s})$.

Therefore,

$$\begin{aligned} \Delta V(\mathbf{s}) &\leq \sum_{i=1}^b N(s_i - s_{i+1}) (V(\mathbf{s} - e_i) - V(\mathbf{s})) + \frac{N\lambda}{k} \left(1 - \frac{1}{N}\right) \left(-\frac{kb}{N(h-1)}\right) \\ &\leq \frac{b}{h-1} (s_1 - s_h) - \frac{b}{h-1} \left(1 - \frac{1}{N}\right) (1 - \beta N^{-\alpha}) \\ &\leq \frac{b}{h-1} \left(1 - \left(\beta N^{-\alpha} + \frac{\log N}{\sqrt{N}}\right)\right) - \left(1 - \frac{1}{N}\right) (1 - \beta N^{-\alpha}), \\ &= \frac{b}{h-1} \left(-\frac{\log N}{\sqrt{N}} + \frac{1}{N} (1 - \beta N^{-\alpha})\right) \\ &\leq -\frac{b}{h-1} \frac{\log N - \frac{1}{\sqrt{N}}}{\sqrt{N}}, \end{aligned} \tag{36}$$

where (36) is due to the fact that $s_1 \leq 1$ and the fact that $s_h \geq \beta N^{-\alpha} + \frac{\log N}{\sqrt{N}}$ following similar arguments as those in Case 1 noting that $\mathcal{T}_1 > \mathcal{T}_2 > B$. When N is sufficiently large, this upper bound becomes

$$\Delta V(\mathbf{s}) \leq -\frac{b}{\sqrt{N}},$$

which completes the proof of the drift bound in Lemma 4.3.

For this Lyapunov function V , under the notation in Lemma A.1, we have that $v_{\max} \leq \frac{kb}{N(h-1)}$ and $f_{\max} \leq \frac{b}{h-1}$. Let $\mathcal{E} = \mathcal{S}$ and $j = \sqrt{N} \log^2 N$. Then by Lemma A.1, the drift bound implies that

$$\begin{aligned} & \mathbb{P} \left\{ V(\mathcal{S}) > B + \frac{2kb \log^2 N}{(h-1)\sqrt{N}} \right\} \\ &= \mathbb{P} \left\{ V(\mathcal{S}) > B + \frac{2kb}{(h-1)N} j \right\} \\ &\leq \left(1 + \frac{h-1}{\sqrt{N}} \right)^{-j} \\ &\leq \left(\left(1 + \frac{1}{\sqrt{N}} \right)^{\sqrt{N}+1} \right)^{-\frac{1}{\sqrt{N}+1} \sqrt{N} \log^2 N} \\ &\leq e^{-\frac{1}{2} \log^2 N}, \end{aligned}$$

where the last inequality holds when N is sufficiently large. This completes the proof. \square

B LEMMAS NEEDED FOR IMPOSSIBILITY RESULTS

B.1 Lemma B.1

LEMMA B.1. *Assume that the system is stable. Then for any $x > 0$,*

$$\mathbb{P} \{S_1 < 1 - x\} \leq \frac{\beta N^{-\alpha}}{x}.$$

PROOF. By work conservation law, it holds that $\mathbb{E}[S_1] = \lambda = 1 - \beta N^{-\alpha}$. Then $\mathbb{E}[1 - S_1] = \beta N^{-\alpha}$. Therefore, by the Markov inequality, for any $x > 0$,

$$\mathbb{P} \{S_1 < 1 - x\} = \mathbb{P} \{1 - S_1 > x\} \leq \frac{\beta N^{-\alpha}}{x}.$$

\square

B.2 Lemma B.2

LEMMA B.2. *Let ℓ be a threshold such that $1 \leq \ell \leq h$ with $h = O(\log k)$. Suppose that an incoming job sees a state \mathbf{s} such that $\sum_{i=1}^{\ell} s_i \geq \ell - x$, where $x = \Omega(hN^{-\alpha})$ and $x = e^{-\Omega(\log N)}$. Consider a Lyapunov function $V_{\ell}(\mathbf{s}) = s_1 + s_2 + \dots + s_{\ell}$. It holds that when N is sufficiently large,*

$$\sum_{s': s \rightarrow s' \text{ due to an arrival}} r_{s \rightarrow s'} (V_{\ell}(s') - V_{\ell}(s)) \leq 2kdx,$$

where $r_{s \rightarrow s'}$ is the transition rate, and $s \rightarrow s'$ due to an arrival means that \mathbf{s} will move to state \mathbf{s}' on the Markov chain only if there is an incoming job.

PROOF. Suppose that an arrival sees a state \mathbf{s} . Given $\sum_{i=1}^{\ell} s_i \geq \ell - x$, we have $s_{\ell} \geq 1 - x$ since $s_i \leq 1$ for all $1 \leq i \leq \ell$. Without loss of generality, we can think of the batch-filling policy as sampling the kd queues one by one. During the sampling, we always choose at most kd servers of length at least ℓ . The probability that all kd sampled servers have length at least ℓ is thus larger or equal to

$$\left(\frac{N(1-x) - kd}{N} \right)^{kd} = \left(1 - \left(x + \frac{kd}{N} \right) \right)^{kd}.$$

Recall that by the assumptions in Theorem 3.3, we have $x = e^{-\Omega(\log N)}$, $kd = o(N^{1-\alpha})$, and thus $x + \frac{kd}{N} > -1$ when N is sufficiently large. Furthermore, applying Bernoulli's Inequality and the assumption that $x = \Omega(hN^{-\alpha})$, it holds

$$\left(1 - \left(x + \frac{kd}{N}\right)\right)^{kd} \geq 1 - kd \left(x + \frac{kd}{N}\right) \geq 1 - 2kdx$$

for a large N . Note that if we put all tasks of this arrival into servers of length at least ℓ , we will not affect the value of $V_\ell(\mathbf{s})$. As a result,

$$\begin{aligned} & \sum_{s': s \rightarrow s' \text{ due to an arrival}} r_{s \rightarrow s'} (V_\ell(\mathbf{s}') - V_\ell(\mathbf{s})) \\ & \leq (1 - 2kdx) \cdot 0 \cdot \frac{\lambda}{k} + 2kdx \cdot k \frac{\lambda}{k} \\ & \leq 2kdx, \end{aligned}$$

which completes the proof. \square

B.3 Lemma B.3

Lemma B.3 is a key in establishing the inductive proof. This lemma relates S_q to S_{q-1} for $3 \leq i \leq h$.

LEMMA B.3. Define $u = 2kd$ and $b_q = u^{q-1} h^q \beta N^{-\alpha}$ for $q \in \mathbb{N}$. Define a sequence a_q , such that $a_1 = 0$, $a_2 = 1$ and $a_q = (q-2)a_{q-1} + 2$ for $q > 2$. For any q with $3 \leq q \leq h$, if

$$\mathbb{P} \{S_1 - S_{q-1} \leq a_{q-1} b_{q-1}\} \geq \left(\frac{h-2}{h}\right)^{q-2} - (q-2)N^{-\log N},$$

then

$$\mathbb{P} \{S_1 - S_q \leq a_q b_q\} \geq \left(\frac{h-2}{h}\right)^{q-1} - (q-1)N^{-\log N}.$$

PROOF. The proof is close to that of Theorem 3.3. Recall that for each $1 \leq \ell \leq h$ and state $\mathbf{s} \in \mathcal{S}$, we define the Lyapunov function

$$V_\ell(\mathbf{s}) = \sum_{i=1}^{\ell} s_i.$$

For q such that $3 \leq q \leq h$, by assumption,

$$\mathbb{P} \{S_1 - S_{q-1} \leq a_{q-1} b_{q-1}\} \geq \left(\frac{h-2}{h}\right)^{q-2} - (q-2)N^{-\log N}.$$

It holds

$$\begin{aligned} & \mathbb{P} \{V_{q-1}(\mathbf{S}) < q-1 - ((q-2)a_{q-1} + 1) b_{q-1}\} \\ & \leq \mathbb{P} \{V_{q-1}(\mathbf{S}) < q-1 - ((q-2)a_{q-1} + 1) b_{q-1}, \\ & \quad S_1 - S_{q-1} \leq a_{q-1} b_{q-1}\} \\ & \quad + \mathbb{P} \{S_1 - S_{q-1} > a_{q-1} b_{q-1}\} \\ & \leq \mathbb{P} \{(q-1)S_1 < q-1 - b_{q-1}\} + 1 - \left(\frac{h-2}{h}\right)^{q-2} \\ & \quad + (q-2)N^{-\log N} \\ & \leq \frac{q-1}{u^{q-2} h^{q-1}} + 1 - \left(\frac{h-2}{h}\right)^{q-2} + (q-2)N^{-\log N}. \end{aligned} \tag{37}$$

The last inequality uses Lemma B.1 and $b_{q-1} = u^{q-2}h^{q-1}\beta N^{-\alpha}$.

Now let $B_{q-1} = q - 1 - ((q - 2)a_{q-1} + 2)b_{q-1}$. We can see that $B_{q-1} = q - 1 - a_q b_{q-1}$. For a state \mathbf{s} such that $V_{q-1}(\mathbf{s}) > B_{q-1}$, it holds

$$\begin{aligned} \Delta V_{q-1}(\mathbf{s}) &= \sum_{s': \mathbf{s} \rightarrow s' \text{ due to an arrival}} r_{\mathbf{s} \rightarrow s'} (V_{q-1}(s') - V_{q-1}(\mathbf{s})) \\ &\quad + \sum_{s': \mathbf{s} \rightarrow s' \text{ due to a departure}} r_{\mathbf{s} \rightarrow s'} (V_{q-1}(s') - V_{q-1}(\mathbf{s})). \end{aligned}$$

Recall that we define $u = 2kd$ and $b_q = u^{q-1}h^q\beta N^{-\alpha}$. As $V_{q-1}(\mathbf{s}) > q - 1 - a_q b_{q-1}$, by Lemma B.2, it holds

$$\begin{aligned} \Delta V_{q-1}(\mathbf{s}) &\leq 2kda_q b_{q-1} - (s_1 - s_q) \\ &= a_q u^{q-1} h^{q-1} \beta N^{-\alpha} - (s_1 - s_q). \end{aligned}$$

Let $\mathbb{P}\{S_1 - S_q \leq a_q b_q\} = p_q$, $\mathcal{E}_{q-1} = \{s \in \mathcal{S} \mid s_1 - s_q > a_q b_q\}$. Then $\mathbb{P}\{S \notin \mathcal{E}_{q-1}\} = p_q$. For a state \mathbf{s} , consider the following two cases.

- $\mathbf{s} \notin \mathcal{E}_{q-1}$, $\Delta V_{q-1}(\mathbf{s}) \leq a_q u^{q-1} h^{q-1} \beta N^{-\alpha} =: \delta$.
- $\mathbf{s} \in \mathcal{E}_{q-1}$. Let $\gamma = -\Delta V_{q-1}(\mathbf{s})$. It holds

$$\gamma \geq a_q u^{q-1} h^{q-1} \beta N^{-\alpha} (h - 1).$$

We then utilize the tail bound, Lemma A.1. Following the definition in Lemma A.1, it is easy to verify that $v_{\max} \leq \frac{k}{N}$, $f_{\max} \leq 1$ for the Lyapunov function $V_{q-1}(\mathbf{s})$. Let

$$j_{q-1} = \left(1 + \frac{N^\alpha}{a_q u^{q-1} h^{q-1} (h - 1) \beta}\right) \log^2 N.$$

Using Lemma A.1,

$$\begin{aligned} &\mathbb{P}\{V_{q-1}(\mathbf{S}) > B_{q-1} + 2v_{\max} j_{q-1}\} \\ &\leq \left(\frac{f_{\max}}{f_{\max} + \gamma}\right)^{j_{q-1}} + \left(\frac{\delta}{\gamma} + 1\right) \mathbb{P}\{S \notin \mathcal{E}_{q-1}\} \\ &\leq \left(\frac{f_{\max}}{f_{\max} + \gamma}\right)^{j_{q-1}} + \frac{h}{h - 1} p_q. \end{aligned}$$

Note that when N is sufficiently large,

$$\left(\frac{f_{\max}}{f_{\max} + \gamma}\right)^{j_{q-1}} \leq e^{-\log^2 N}.$$

Besides, we assume that $0 < \alpha < 0.5$, $k = e^{o(\sqrt{\log N})}$ and $h = O(\log k)$. As a result, for a large N ,

$$\begin{aligned} &\mathbb{P}\{V_{q-1}(\mathbf{S}) \geq q - 1 - ((q - 2)a_{q-1} + 1)b_{q-1}\} \\ &\leq \mathbb{P}\{V_{q-1}(\mathbf{S}) > B + 2v_{\max} j_{q-1}\} \\ &\leq e^{-\log^2 N} + \frac{h}{h - 1} p_q. \end{aligned}$$

Together with Eq.(37), we have

$$\begin{aligned} &\left(\frac{h - 2}{h}\right)^{q-2} - \frac{q - 1}{u^{q-2} h^{q-1}} - (q - 2)N^{-\log N} \\ &\leq \mathbb{P}\{V_{q-1}(\mathbf{S}) > q - 1 - ((q - 2)a_{q-1} + 1)b_{q-1}\} \\ &\leq e^{-\log^2 N} + \frac{h}{h - 1} p_q. \end{aligned}$$

We can conclude that for a large N ,

$$\mathbb{P} \{S_1 - S_q \leq a_q b_q\} = p_q \geq \left(\frac{h-2}{h}\right)^{q-1} - (q-1)N^{-\log N},$$

which completes the proof. \square

B.4 Lemma B.4

Lemma B.4 complements the probability bound in Lemma 4.2. Recall that FILL_h denotes the event that all the k tasks of an incoming job are assigned to queueing positions below a threshold h . Lemma B.4 gives a condition on the total queue length for FILL_h to happen with low probability.

LEMMA B.4. *Suppose an incoming job sees a state s such that $\sum_{i=1}^h s_i > h - \frac{1}{3d}$. Then when N is sufficiently large,*

$$\mathbb{P} \{\text{FILL}_h\} = o(1).$$

PROOF. We use a similar argument as the proof of Lemma 4.2. Suppose that an arrival sees a state s . By assumption, it holds

$$\sum_{i=1}^h s_i \geq h - \frac{1}{3d}.$$

Let X_1, \dots, X_{kd} be the numbers of places below h in each sampled server. The goal is to show

$$\mathbb{P} \{\text{FILL}_h\} = \mathbb{P} \left\{ \sum_{i=1}^{kd} X_i \geq k \right\} = o(1)$$

when N is large enough.

We could see that for each integer x such that $1 \leq x \leq h$, $\mathbb{P}\{X_i = x\} = s_{h-x} - s_{h-x+1}$, and $\mathbb{P}\{X_i = 0\} = s_h$. Since we are sampling without replacement, X_1, \dots, X_{kd} are not independent. But still, utilizing a result of Hoeffding [20, Theorem 4], we have $\mathbb{E} \left[f \left(\sum_{i=1}^{kd} X_i \right) \right] \leq \mathbb{E} \left[f \left(\sum_{i=1}^{kd} Y_i \right) \right]$ for any continuous and convex function $f(\cdot)$, where Y_1, \dots, Y_{kd} are i.i.d. and follow the same distribution as X_1 . Take $f(\cdot)$ to be $f(x) = e^{tx}$ where t is some positive value.

It then holds

$$\begin{aligned} \mathbb{P} \{\text{FILL}_h\} &= \mathbb{P} \left\{ \sum_{i=1}^{kd} X_i \geq k \right\} \\ &= \mathbb{P} \left\{ e^{t \sum_{i=1}^{kd} X_i} \geq e^{tk} \right\} \\ &\leq e^{-tk} \prod_{i=1}^{kd} \mathbb{E} [e^{tY_i}] \\ &= e^{-tk} \prod_{i=1}^{kd} \left(1 + \sum_{j=1}^h \left(e^{t(h-j+1)} - 1 \right) \right). \end{aligned}$$

Since for all $x > 0$, $1 + x \leq e^x$, we can further have

$$\mathbb{P} \{\text{FILL}_h\} \leq e^{-tk} \exp \left(kd \sum_{j=1}^h \left(e^{t(h-j+1)} - 1 \right) (s_{j-1} - s_j) \right). \quad (38)$$

Rearranging the sum in (38), we get

$$\begin{aligned}
& \sum_{j=1}^h \left(e^{t(h-j+1)} - 1 \right) (s_{j-1} - s_j) \\
&= e^{th} - \sum_{j=1}^h s_j \left(e^{t(h-j+1)} - e^{t(h-j)} \right) \\
&= e^{th} - (e^t - 1) \sum_{j=1}^h s_j e^{t(h-j)}.
\end{aligned} \tag{39}$$

Recall that $\sum_{j=1}^h s_j \geq h - \frac{1}{3d}$, and $1 \geq s_1 \geq s_2 \geq \dots \geq s_h \geq 0$. Eq. (39) is maximized when $s_1 = s_2 = \dots = s_h = 1 - \frac{1}{3dh}$ and thus,

$$(39) \leq (e^{th} - 1) \frac{1}{3dh}.$$

Plug it into Inequality (38),

$$\mathbb{P}\{\text{FILL}_h\} \leq \min_{t>0} \exp \left(k \left(-t + \frac{e^{th} - 1}{3h} \right) \right).$$

Pick $t = \frac{\ln 3}{h}$. It holds

$$\mathbb{P}\{\text{FILL}_h\} \leq \exp \left(\frac{k}{3h} (-3 \ln 3 + 2) \right).$$

By the assumption that $\frac{k}{h} = \omega(1)$, we could conclude that

$$\mathbb{P}\{\text{FILL}_h\} = o(1)$$

when N is sufficiently large. □

C PROOF OF THEOREM 6.1

PROOF. Let \mathcal{I} be the event that all the tasks of an incoming job are assigned to idle servers in steady state. Then what we need to show is $\mathbb{P}\{\mathcal{I}\} \leq 0.5$.

From the stability of batch-filling [45] and the Little's law, it holds $\mathbb{E}S_1 = \lambda$. For a job arrival of k tasks, in order to schedule every task to an idle server, batch-filling needs to find at least k idle servers. Suppose batch-filling probes kd servers with state X_1, \dots, X_{kd} where X_i is a 0-1 random variables indicating whether the sampled i th server is idle. Then

$$\mathbb{P}\{\mathcal{I}\} = \mathbb{P}\{X_1 + \dots + X_{kd} \geq k\}.$$

Notice that $\mathbb{E}[X_1 + \dots + X_{kd}] = kd(1 - \lambda)$ by the linearity of expectations. If $d \leq \frac{1}{2(1-\lambda)}$, this expectation is upper bounded by $\frac{k}{2}$. Therefore,

$$\mathbb{P}\{\mathcal{I}\} = \mathbb{P}\{X_1 + \dots + X_{kd} \geq k\} \leq \frac{\mathbb{E}[X_1 + \dots + X_{kd}]}{k} \leq 0.5.$$

□

D MORE DETAILS ON SIMULATIONS

D.1 Probe Ratios

In the simulations, we need to adjust the definition of probe ratio a little. Let $D_i = \lfloor \min(N, kd_i) \rfloor$ for $1 \leq i \leq 4$. Then D_i is the true number of probes used in batch-filling for each job. When N is small, D_i may be equal to N . In this case, we adjust the value of d_i as $\frac{D_i}{k}$, which is the true expected probe ratio of each task. The exact value of d_i is shown in Table 2.

N	d_1	d_2	d_3	d_4
32	2.7	2.1	2.7	2.7
64	3.8	2.2	3.8	3.8
128	5.6	2.2	4.7	5.6
256	7.6	2.3	5.1	8.2
512	9.3	2.4	5.5	9.4
1024	11.3	2.5	5.9	10.8
2048	13.7	2.6	6.3	12.4
4096	16.6	2.7	6.8	14.4
8192	20.2	2.8	7.3	16.7
16384	24.5	3.0	7.9	19.4
32768	29.9	3.1	8.5	22.6
65536	36.5	3.2	9.3	26.5

Table 2. Probe Ratios for Different Scales of System

D.2 Numerical Values for Figures 3 and 4

We give the numerical values and standard deviations for Figures 3 and 4 in Tables 3 and 4, respectively.

N	d_1	d_2	d_3	d_4
32	$0.23(\pm 3.8 \times 10^{-4})$	$0.27(\pm 3.2 \times 10^{-4})$	$0.23(\pm 3.8 \times 10^{-4})$	$0.23(\pm 3.8 \times 10^{-4})$
64	$0.20(\pm 4.5 \times 10^{-4})$	$0.30(\pm 3.0 \times 10^{-4})$	$0.20(\pm 4.5 \times 10^{-4})$	$0.20(\pm 4.5 \times 10^{-4})$
128	$0.17(\pm 2.4 \times 10^{-4})$	$0.33(\pm 9.9 \times 10^{-4})$	$0.18(\pm 3.6 \times 10^{-4})$	$0.17(\pm 2.4 \times 10^{-4})$
256	$0.14(\pm 3.7 \times 10^{-4})$	$0.35(\pm 6.6 \times 10^{-4})$	$0.19(\pm 1.0 \times 10^{-4})$	$0.14(\pm 1.7 \times 10^{-4})$
512	$0.13(\pm 6.2 \times 10^{-4})$	$0.37(\pm 7.8 \times 10^{-4})$	$0.19(\pm 3.6 \times 10^{-4})$	$0.13(\pm 1.1 \times 10^{-4})$
1024	$0.12(\pm 1.5 \times 10^{-4})$	$0.40(\pm 1.8 \times 10^{-4})$	$0.20(\pm 3.8 \times 10^{-4})$	$0.12(\pm 2.3 \times 10^{-5})$
2048	$0.10(\pm 2.1 \times 10^{-4})$	$0.41(\pm 3.0 \times 10^{-4})$	$0.21(\pm 4.3 \times 10^{-4})$	$0.11(\pm 1.1 \times 10^{-4})$
4096	$0.09(\pm 6.4 \times 10^{-4})$	$0.43(\pm 1.6 \times 10^{-3})$	$0.24(\pm 4.8 \times 10^{-4})$	$0.10(\pm 2.9 \times 10^{-4})$
8192	$0.08(\pm 2.0 \times 10^{-4})$	$0.45(\pm 8.3 \times 10^{-4})$	$0.26(\pm 1.9 \times 10^{-4})$	$0.10(\pm 4.7 \times 10^{-4})$
16384	$0.07(\pm 2.7 \times 10^{-4})$	$0.48(\pm 6.7 \times 10^{-4})$	$0.28(\pm 5.8 \times 10^{-4})$	$0.09(\pm 4.1 \times 10^{-4})$
32768	$0.05(\pm 2.5 \times 10^{-4})$	$0.51(\pm 1.1 \times 10^{-3})$	$0.30(\pm 2.1 \times 10^{-4})$	$0.08(\pm 2.8 \times 10^{-4})$
65536	$0.05(\pm 2.2 \times 10^{-4})$	$0.54(\pm 5.3 \times 10^{-4})$	$0.31(\pm 2.2 \times 10^{-4})$	$0.10(\pm 4.7 \times 10^{-4})$

Table 3. Values of $\frac{\mathbb{E}[T-T^*]}{\mathbb{E}[T^*]}$ in Figure 3

D.3 Delay Scaling when $k = \lfloor \sqrt{N} \rfloor$

In this section, we provide simulation results for the setting where $k = \lfloor \sqrt{N} \rfloor$. The scalings of probe ratios are the same as in Section 7.1. The results are demonstrated in Figure 5, and the numerical values and standard deviations are given in Table 5.

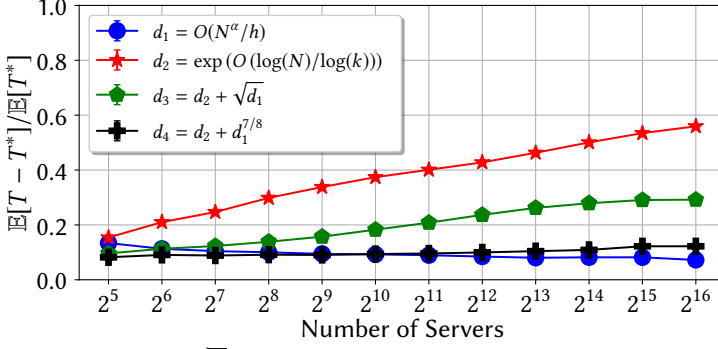


Fig. 5. Queueing delays when $k = \lfloor \sqrt{N} \rfloor$ under different probe ratios: d_1 is sufficient for convergence to zero queueing delay; $d_1 > d_4 > d_3 > d_2$.

Received August 2020; revised September 2020; accepted October 2020

N	Exponential	Hyper-Exponential	Bounded Pareto	S & X
32	0.23($\pm 3.8 \times 10^{-4}$)	0.84($\pm 2.7 \times 10^{-3}$)	0.49($\pm 1.4 \times 10^{-3}$)	0.55($\pm 8.1 \times 10^{-4}$)
64	0.20($\pm 4.5 \times 10^{-4}$)	0.73($\pm 1.8 \times 10^{-3}$)	0.46($\pm 1.6 \times 10^{-3}$)	0.53($\pm 1.2 \times 10^{-3}$)
128	0.17($\pm 2.4 \times 10^{-4}$)	0.61($\pm 8.3 \times 10^{-4}$)	0.41($\pm 6.0 \times 10^{-4}$)	0.48($\pm 8.5 \times 10^{-4}$)
256	0.14($\pm 3.7 \times 10^{-4}$)	0.50($\pm 9.5 \times 10^{-4}$)	0.36($\pm 1.2 \times 10^{-3}$)	0.44($\pm 1.8 \times 10^{-3}$)
512	0.13($\pm 6.2 \times 10^{-4}$)	0.42($\pm 4.1 \times 10^{-4}$)	0.31($\pm 2.7 \times 10^{-4}$)	0.40($\pm 4.3 \times 10^{-4}$)
1024	0.12($\pm 1.5 \times 10^{-4}$)	0.34($\pm 6.3 \times 10^{-4}$)	0.27($\pm 2.7 \times 10^{-4}$)	0.34($\pm 5.3 \times 10^{-4}$)
2048	0.10($\pm 2.1 \times 10^{-4}$)	0.27($\pm 4.3 \times 10^{-4}$)	0.23($\pm 4.0 \times 10^{-4}$)	0.28($\pm 9.3 \times 10^{-4}$)
4096	0.09($\pm 6.4 \times 10^{-4}$)	0.19($\pm 9.2 \times 10^{-3}$)	0.19($\pm 2.7 \times 10^{-4}$)	0.23($\pm 8.6 \times 10^{-4}$)
8192	0.08($\pm 2.0 \times 10^{-4}$)	0.12($\pm 4.7 \times 10^{-4}$)	0.13($\pm 4.6 \times 10^{-4}$)	0.18($\pm 7.1 \times 10^{-4}$)
16384	0.07($\pm 2.7 \times 10^{-4}$)	0.07($\pm 9.0 \times 10^{-4}$)	0.10($\pm 4.0 \times 10^{-4}$)	0.12($\pm 5.2 \times 10^{-4}$)
32768	0.05($\pm 2.5 \times 10^{-4}$)	0.03($\pm 4.0 \times 10^{-3}$)	0.03($\pm 2.0 \times 10^{-4}$)	0.08($\pm 3.0 \times 10^{-4}$)
65536	0.05($\pm 2.2 \times 10^{-4}$)	0.01($\pm 2.3 \times 10^{-4}$)	0.02($\pm 1.4 \times 10^{-4}$)	0.06($\pm 1.0 \times 10^{-3}$)

Table 4. Values of $\frac{\mathbb{E}[T - T^*]}{\mathbb{E}[T^*]}$ in Figure 4

N	d_1	d_2	d_3	d_4
32	$0.13(\pm 5.2 \times 10^{-4})$	$0.15(\pm 5.2 \times 10^{-4})$	$0.10(\pm 2.0 \times 10^{-4})$	$0.08(\pm 2.9 \times 10^{-4})$
64	$0.11(\pm 4.2 \times 10^{-4})$	$0.21(\pm 8.1 \times 10^{-4})$	$0.11(\pm 4.1 \times 10^{-4})$	$0.09(\pm 4.1 \times 10^{-4})$
128	$0.10(\pm 7.6 \times 10^{-4})$	$0.25(\pm 6.5 \times 10^{-4})$	$0.12(\pm 4.8 \times 10^{-4})$	$0.09(\pm 7.2 \times 10^{-4})$
256	$0.10(\pm 3.4 \times 10^{-4})$	$0.30(\pm 4.7 \times 10^{-4})$	$0.14(\pm 7.5 \times 10^{-4})$	$0.09(\pm 3.0 \times 10^{-4})$
512	$0.09(\pm 3.8 \times 10^{-4})$	$0.34(\pm 2.6 \times 10^{-4})$	$0.16(\pm 2.7 \times 10^{-4})$	$0.09(\pm 2.7 \times 10^{-4})$
1024	$0.09(\pm 1.4 \times 10^{-4})$	$0.37(\pm 8.2 \times 10^{-4})$	$0.18(\pm 4.9 \times 10^{-4})$	$0.09(\pm 9.5 \times 10^{-5})$
2048	$0.09(\pm 2.9 \times 10^{-4})$	$0.4(\pm 2.1 \times 10^{-4})$	$0.21(\pm 4.0 \times 10^{-4})$	$0.10(\pm 1.9 \times 10^{-4})$
4096	$0.08(\pm 2.6 \times 10^{-4})$	$0.43(\pm 9.8 \times 10^{-4})$	$0.24(\pm 5.0 \times 10^{-4})$	$0.10(\pm 2.2 \times 10^{-4})$
8192	$0.08(\pm 3.5 \times 10^{-4})$	$0.46(\pm 2.2 \times 10^{-4})$	$0.26(\pm 1.3 \times 10^{-4})$	$0.10(\pm 3.9 \times 10^{-4})$
16384	$0.08(\pm 5.3 \times 10^{-4})$	$0.50(\pm 7.0 \times 10^{-4})$	$0.28(\pm 1.7 \times 10^{-4})$	$0.11(\pm 3.1 \times 10^{-4})$
32768	$0.08(\pm 4.9 \times 10^{-4})$	$0.53(\pm 1.7 \times 10^{-4})$	$0.29(\pm 1.5 \times 10^{-4})$	$0.12(\pm 1.9 \times 10^{-4})$
65536	$0.07(\pm 3.1 \times 10^{-4})$	$0.56(\pm 9.8 \times 10^{-4})$	$0.29(\pm 1.6 \times 10^{-4})$	$0.12(\pm 3.8 \times 10^{-4})$

Table 5. Values of $\frac{\mathbb{E}[T-T^*]}{\mathbb{E}[T^*]}$ in Figure 5