Delay-optimal policies in partial fork-join systems with redundancy and random slowdowns

DELAY-OPTIMAL POLICIES IN PARTIAL FORK-JOIN SYSTEMS WITH REDUNDANCY AND RANDOM SLOWDOWNS

By Martin Zubeldia

Eindhoven University of Technology and University of Amsterdam

We consider a large distributed service system consisting of n homogeneous servers with infinite capacity FIFO queues. Jobs arrive as a Poisson process of rate $\lambda n/k_n$ (for some positive constant λ and integer k_n). Each incoming job consists of k_n identical tasks that can be executed in parallel, and that can be encoded into at least k_n "replicas" of the same size (by introducing redundancy) so that the job is considered to be completed when *any* k_n replicas associated with it finish their service. Moreover, we assume that servers can experience random slowdowns in their processing rate so that the service time of a replica is the product of its size and a random slowdown.

First, we assume that the server slowdowns are shifted exponential and independent of the replica sizes. In this setting we show that the delay of a typical job is asymptotically minimized (as $n \to \infty$) when the number of replicas per task is a constant that only depends on the arrival rate λ, and on the expected slowdown of servers.

Second, we introduce a new model for the server slowdowns in which larger tasks experience less variable slowdowns than smaller tasks. In this setting we show that, under the class of policies where all replicas start their service at the same time, the delay of a typical job is asymptotically minimized (as $n \to \infty$) when the number of replicas per task is made to depend on the actual size of the tasks being replicated, with smaller tasks being replicated more than larger tasks.

CONTENTS

1. Introduction. Consider a distributed service system consisting of a large number of servers operating in parallel, where each server can experience random slowdowns in its processing rate. Each incoming job consists of k identical tasks that can be executed in parallel, and that can be encoded into at least k "replicas" of the same size (by introducing redundancy) so that the job is considered to be completed when *any* k replicas associated with it finish their service. For the case of $k = 1$, this corresponds to simply creating copies of the job, dispatching them to different servers, and waiting for the first one to finish. For the general case with $k > 1$, our model is a generalization of the classic fork-join model (which assumes $k = n$), and it is motivated by the following applications:

(i) **Data file retrieval with redundancy and coding:** Consider a setting where a user wants to retrieve a large file, which can be downloaded from a large set of servers. In order to shorten the download time, the following scheme can be used [14]. The file is split into k pieces of equal size, and then the k pieces are encoded into $r \geq k$ pieces of the same size, in a way that allows the original file to be recovered from any k pieces. The user starts to download the r encoded pieces from r different servers, waits for the first k to finish, and then cancels the other downloads. In this setting, the slowdowns in the download come from the congestion in the route between the user that requests the file and the servers from which the pieces are downloaded, as well as from congestion at the server themselves. This encoding scheme was shown to be better than replicating each of the pieces in [9].

(ii) **Approximate distributed computing:** Consider a setting where a user wants to compute k gradient estimates in a Parallelized Stochastic Gradient Descent method, executed by a large server farm with $n \gg k$ servers [8]. In order to reduce processing time, we attempt to compute $r \geq k$ gradient estimates and we stop when any k computations are finished. In this setting, the slowdowns come from reductions in server processing power due to background processes or similar issues.

In general, there are many sources of randomness for the service time of a task. These include:

(i) The intrinsic task size variability.
(ii) Slowdowns in the local processing rate of the server due to exogenous interferences (such as background processes).
(iii) Network congestion that interferes with communication.

The intrinsic task size variability is always a source of randomness in the service times (even when there are no slowdowns), and it is captured in almost every model in the literature, including the two models that we will work with. However, not all other sources of randomness are relevant for all applications. For instance, network congestion interfering with communication can create significant slowdowns for the retrieval of data from a server, whereas the reduction in the local processing power of a server can barely affect it. This behavior is more accurately reflected by the independent exponential slowdowns model introduced in Section 3. On the other hand, random reductions in the processing power of local servers have a greater impact on the slowdowns in distributed computing than network congestion. This behavior is more accurately reflected by the size-based slowdowns

model introduced in Section 4.

Our objective is to understand the best possible performance of such systems and to propose near-optimal policies, with emphasis on the asymptotic regime when n is large. In particular, our performance metric is the **delay** of a typical job, i.e., the time between a job arrives to the system, and the time when k replicas associated with it finish their services. The delay can be decomposed as the sum of the **service time**, i.e., the total amount of time that at least one replica associated with the job is in service, and the **waiting time**, i.e., the total amount of time that no replica associated with the job is in service.

A control policy for such systems must specify how many redundant tasks (called **replicas** from now on) to create, when to create them (their creation can be staggered in time), and to which servers to send them. Furthermore, replicas can also be prematurely cancelled. All of these decisions have the potential of reducing the delay of a typical job.

1. For example, suppose that we send r replicas to different servers, but as soon as any k replicas associated with the same job start their service, all other replicas associated with the same job are cancelled. In this case, the k replicas in service correspond to the k-th shortest queues out of the original r. This effectively reduces the queueing delay of replicas, and thus the queueing delay of the job. Furthermore, since the other replicas never start service, they do not consume any resources and do not affect the delays of other jobs.
2. Even if all replicas experience zero queueing delays, the creation of more replicas can reduce the delay of the job. This is because replicas processed by servers with higher processing rates (i.e., that experience less severe slowdowns) finish their service earlier. Thus, if $r > k$ replicas start their service at the same time, the delay of the job will be the k-th smallest out of the r service times of the replicas associated with it.
3. While the creation of more replicas always reduces the delay of an individual job, too many replicas can lead to instability. Furthermore, even before the system becomes unstable, the cost of the increased congestion may overshadow the reduction in service times.

The combination of having multiple tasks per job, the ability to create replicas, and the random slowdowns in the service rates, make replication-based systems difficult to analyze, even under simple control policies.

1.1. *Previous work.* Most of the prior theoretical work on parallel service systems with redundancy has been made for the case of a single task per job (i.e., the case $k_n = 1$), and under the (unrealistic) *independent runtimes model*, which stipulates that the times that replicas require to complete their service are i.i.d., regardless of whether they are associated with the same job or not. Using this model, a body of work has focused on characterizing optimal replication policies based on the log-concavity of the complementary cdf of the service times [15, 16, 26, 29]. In particular, it has been shown that if the complementary cdf of the service time distribution of a replica is log-convex, then the minimum delay is achieved when replicas are sent to all the servers. Furthermore, in [12, 31, 32, 33] the authors explore the tradeoffs between the delays and the resources utilized, for several classes of service time distributions (mainly discrete and log-convex). The stability and performance for difference scheduling policies at the queues was studied in [3, 13].

On the other hand, there is some recent work [11] that introduces a more realistic model where service times of replicas associated with the same job are correlated. Policies under this model are significantly harder to analyze, and the available theoretical results are limited. In [24, 14], the authors obtain results about the performance of particular policies. Furthermore, in [10] the authors develop policies that result in fair delays for multiple classes of jobs, under replication constraints.

Results from different settings can also cast a light on delay-optimal replication schemes. For example, in [6] the authors analyse a discrete-time parallel service system, where the service times of replicas are i.i.d. and geometrically distributed, and where replicas can be created and deleted after each time slot, so that the number of servers working on a job can change from slot to slot. In this setting, it has been shown that the delay is minimized when all servers are used all of the time, and the number of replicas associated with each job are all equal (or differ by at most 1). Furthermore, in [5] the authors analyse a different parallel service system where, instead of replication, jobs are amenable to parallel processing, with a sublinear improvement in processing rate. For the case of exponentially distributed jobs it was shown that if the number of servers processing a job can be changed at any point in time, then the delay is minimized when all servers are used all of the time, and the number of servers that process each job are all equal (or differ by at most 1).

Finally, for the more general case where $k_n > 1$, our model is a generalization of the classic fork-join model, where the number of tasks per job is equal to the number of servers (i.e., where $k_n = n$). In this setting, tight

characterizations of the delay are only known for the two-server case (see [30] for a detailed survey). Although there are no tight delay characterizations for fork-join models, there are several asymptotic and non-asymptotic bounds [17, 18, 23, 25, 27, 34], with different levels of tightness.

1.2. *Our contribution.* We consider a broad family of control policies, which includes most policies considered in the literature, and work towards characterizing the achievable delay performance of jobs under two different models for the slowdowns.

1. We consider the $S\&X$ model, first introduced in [11], which assumes that the slowdowns are independent from the task sizes. This is a plausible model of the slowdowns which fluctuate on a time scale that is slower than the typical delay. Moreover, we assume that the slowdowns are exponential, which are observed, for example, in the download times from Amazon servers [7, 20]. Under this model, our first contribution is a universal lower bound for the expected delay of a typical job under any control policy, which provides a benchmark for any practical policy. Surprisingly, this lower bound is independent of the task size distribution, of the distribution of the inter-arrival times, and of the number of servers. Our second contribution is the introduction of simple control policies that asymptotically achieve the lower bound, under minor technical conditions. For these asymptotically optimal policies, the number of replicas created per task (i.e., the quantity r/k) is independent of the task sizes, and of the number of tasks.
2. We also consider a new Size-based slowdown model, under which the distribution of the slowdowns is a function of the task sizes. This reflects the fact that longer replicas should "average out" the slowdowns and experience less variability in their service times than short job. This is more accurate for modeling slowdowns that are in the same time-scale as the delays (e.g., the slowdowns in distributed computing). Under this model, we consider policies where all replicas associated with the same job start their service at the same time. For policies that are optimal under this restriction, we show that the number of replicas per task is nonincreasing in the size of the task. This is consistent with current practice, but to the best of our knowledge this is the first tractable model that justifies this practice.

1.3. *Outline of the paper.* The rest of the paper is organized as follows. In Section 2 we introduce the general modeling assumptions, and the policies that are considered throughout the paper. In Section 3 we introduce the first

model for the slowdowns, and the main results for that model. In Section 4 we introduce a new sized-based model for the slowdowns, and the main results for this new model. Finally, in Section 5 we present our conclusions and suggestions for future work.

2. Modeling assumptions and performance metrics. Throughout this paper we consider a system consisting of n homogeneous servers, which can experience random slowdowns in their processing rates, and where each server is associated with an infinite capacity FIFO queue. We assume that jobs arrive to the system as a Poisson process of rate $\lambda n/k_n$, for some fixed $\lambda > 0$ and some positive integer $k_n \leq n$. Each job consists of k_n tasks of the same (albeit random) size. Task sizes are i.i.d. across different jobs, and have unit mean. Furthermore, we assume that we can encode the k_n tasks into any number of at least k_n replicas (of the same size as the tasks) such that a job is finished when *any* k_n replicas associated with it finish their service (at which point all remaining replicas associated with the same job are immediately removed from the system).

Service time variability: Let X_j be the size of the replicas associated with the j-th job. As mentioned earlier, the $\{X_j\}_{j\geq 1}$ are i.i.d. The number of replicas created and associated with each job can be random, as it depends on the control policy, congestion, and other factors. Let $S_{j,r}$ be the slowdown that would be experienced by the r-th replica associated with the j-th job, if such a replica were to start its service. In that case, we assume that the time required for the r-th replica associated with the j-th job to finish its service is equal to $X_j(1 + S_{j,r})$. Moreover, we assume the following.

Assumption 2.1.

(i) There exists a family of cumulative distribution functions $\{F_x : x \geq 0\}$ such that, for all $x \geq 0$, and for every $j \geq 1$ and $R \geq 1$, we have

$$\mathbb{P}\left(\bigcap_{r=1}^{R}\{S_{j,r} \leq s_r\} \,\middle|\, X_j = x\right) = \prod_{r=1}^{R} F_x(s_r),$$

for all $s_1, \ldots, s_R \geq 0$.

(ii) The random sequences of slowdowns $\big\{(S_{j,r} : r \geq 1)\big\}_{j\geq 1}$ are independent, i.e., for every $J \geq 1$, we have

$$\mathbb{P}\left(\bigcap_{j=1}^{J}\Big\{(S_{j,r} : r \geq 1) \in A_j\Big\}\right) = \prod_{j=1}^{J}\mathbb{P}\big((S_{j,r} : r \geq 1) \in A_j\big),$$

for all measurable sets $A_1, \ldots, A_J$.

REMARK 2.1. The first part of Assumption 2.1 asserts that, conditioned on the task size, the slowdowns experienced by different replicas associated with the same job are independent and identically distributed, and that the conditional distribution of the slowdowns is the same across different jobs. Combining this with the fact that replicas' sizes are i.i.d., we get that the random sequences of slowdowns $\big\{(S_{j,r} : r \geq 1)\big\}_{j \geq 1}$ are identically distributed, i.e., that

$$(S_{j,r} : r \geq 1) \stackrel{d}{=} (S_{j',r} : r \geq 1),$$

for all $j, j' \geq 1$. Further combining this with the second part of Assumption 2.1, we conclude that the random sequences of slowdowns $\big\{(S_{j,r} : r \geq 1)\big\}_{j \geq 1}$ are i.i.d.

In the two models that we consider, there will be different additional assumptions on the family of cumulative distribution functions $\{F_x : x \geq 0\}$.

2.1. *Admissible control policies.* In this subsection, we introduce a broad family of control policies for our system. In order to do this, we start by introducing the extended queue state process $\mathcal{Q}_n(\cdot)$. The extended queue state $\mathcal{Q}_n(t)$ at time t is a set, where each element corresponds to a job in the system. In particular,

$$\Big(x, u, \big\{d_1, \ldots, d_k\big\}, \big\{(n_1, y_1, e_1), \ldots, (n_r, y_r, e_r)\big\}\Big) \in \mathcal{Q}_n(t)$$

if at time t there is a job in the system that satisfies the following:

(i) the tasks associated with the job have size x,
(ii) the job arrived u units of time ago
(iii) k replicas associated with the job have already finished their service, $d_1, \ldots, d_k$ units of time after the arrival of the job,
(iv) the job has a total of r replicas associated with it still in the system, in queues $n_1, \ldots, n_r$, which
    1. are in positions $y_1, \ldots, y_r$ in their respective queues (with the convention that if a replica is in service then it is in position 0),
    2. have elapsed service times equal to $e_1, \ldots, e_r$.

For example, if at time t there is a job in the system with tasks of size x, which arrived u units of time ago, and no replicas were created yet, we have $(x, u, \emptyset, \emptyset) \in \mathcal{Q}_n(t)$.

The evolution of the process $\mathcal{Q}_n(\cdot)$ is partly driven by intrinsic features of our queueing model, such as the arrival process, the size of incoming tasks, the slowdown of the replicas, the FIFO queues, and the fact that each job leaves the system (together with all of the replicas associated with it) at the moment that k_n of its replicas finish their service. On the other hand, the control policy determines:

(i) when to create new replicas for jobs in the system,
(ii) where to dispatch the newly created replicas,
(iii) when to cancel replicas in the system.

In general, we consider policies that use the current extended queue state to make these decisions. This is formalized in the following assumption.

ASSUMPTION 2.2 (Markovianity). The extended queue state process $\mathcal{Q}_n(\cdot)$ is Markov.

Note that this assumption implies that the control policies considered are completely characterized by the Markov kernel of the corresponding extended queue state process $\mathcal{Q}_n(\cdot)$.

Moreover, the extended queue state process $\mathcal{Q}_n(\cdot)$ is rich enough to allow for policies that take into account task size, time since the last replica associated with a job finished its service, and more, in order to decide when to create and cancel replicas.

We now introduce an assumption that restricts when replicas can be cancelled.

ASSUMPTION 2.3 (No late cancellations). Replicas with positive elapsed times cannot be cancelled (unless k_n of them have finished their service).

This assumption is introduced to keep the policies tractable, but it can lead to a degradation of performance in some cases. For example, suppose that the expected remaining service time of a replica with positive elapsed time is larger than the expected service time of a new replica. In that case, it is better to cancel the preexisting replica and replace it with a new one.

On the other hand, when the distribution of the slowdowns has a non-decreasing hazard rate, the expected remaining service time of a replica with positive elapsed time is always smaller than the expected service time of a new replica. In that case, it is never a good idea to cancel a replica that is being processed to replace it with a new one. Thus, Assumption 2.3 is not very restrictive when the slowdowns have non-decreasing hazard rates (as

will be the case in Section 3).

Moreover, Assumption 2.3 implies the following.

LEMMA 2.1. *Under Assumption 2.3, at most* $n+k_n-1$ *replicas associated with the same job are ever in service.*

PROOF. Suppose that at least $n + k_n$ replicas associated with the same job are eventually in service. Since there can only be at most n replicas in service at the same time (because we have only n FIFO queues), then in order for the $(n + k_n)$-th replica to start its service, at least k_n replicas have to have finished their services before. This is a contradiction, because all replicas associated with the same job leave the system as soon as k_n replicas associated with it finish their services. □

We are now ready to define the admissible control policies that will be used throughout this paper.

DEFINITION 2.1 (Admissible policies). We say that a control policy is *admissible* if the corresponding extended queue state process $\mathcal{Q}_n(\cdot)$ satisfies assumptions 2.2 and 2.3.

REMARK 2.2. Note that a decision maker implementing the policies introduced in this section has full knowledge of the state of all queues in the system, including the size of all tasks. However, the decision maker has no information about the slowdowns of the servers, other than what can be inferred from the state of the queues.

2.2. *Stability and performance metric.* We say that an admissible policy is **stable** if the corresponding process $\mathcal{Q}_n(\cdot)$ admits a unique invariant probability measure. For stable policies, the performance metric of interest is the **expected delay** of a typical job, i.e., the steady-state expectation of the time between the moment a job arrives to the system, and the moment that k_n replicas associated with it finish their service.

The delay of a typical job (denoted by W_n) can be decomposed as the sum of two components. The first component is the **service time** (denoted by W_n^s), and it is defined as the total amount of time that at least one replica associated with the job is in service. The second component is the **queueing delay** (denoted by W_n^q), and it is defined as the total amount of time between the arrival and the departure of the job that no replica associated with it is in service.

REMARK 2.3. Note that, since k_n replicas associated with a job need to finish their service before the job leaves the system, and since the replicas can be staggered in time, a job can be in service intermittently over time. In particular, while a job is in the system, it incurs service time when at least one of the replicas associated with it is in service. The rest of the time it is incurring queueing delay. This is akin to the service time and queueing delay of a job in a preemptive queue, where a job can start and resume its service many times.

3. Independent exponential slowdowns. In this section, we explore the stability and delay performance of admissible policies under the assumption that slowdowns are exponential and independent from the replica size. In particular, in Subsection 3.1 we obtain necessary and sufficient conditions in the parameters of the system for the existence of stable policies, and we obtain a universal lower bound on the expected delay of any admissible policy. Moreover, in Subsection 3.2 we introduce a pair of admissible policies and show that they are asymptotically delay-optimal under certain conditions on the moments of the task sizes.

Throughout this section, we assume the following.

ASSUMPTION 3.1. For all $j, r \geq 1$, we have

$$\mathbb{P}\big(S_{j,r} \leq s\big) = 1 - e^{-\mu s},$$

for all $s \geq 0$.

Note that combining Assumption 3.1 with Assumption 2.1 we get that, for all $j \geq 1$, the slowdowns $\{S_{j,r}\}_{r\geq 1}$ are i.i.d. exponential random variables with mean $1/\mu$, which are also independent from the replicas' size X_j. This corresponds to the $S\&X$ model introduced in [11], for the special case where slowdowns are exponential. On the practical side, exponential slowdowns are observed, for example, in the download times from Amazon servers [7, 20].

In what follows, we present a systematic approach to the analysis and design of policies, which culminates with the introduction and analysis of asymptotically optimal policies.

3.1. *Delay lower bound.* In this subsection, we first obtain a necessary and sufficient condition on the parameters of the system for the existence of stable admissible policies. Then, we establish a lower bound on the expected delay of a typical job for any stable admissible policy.

THEOREM 3.1. *Under Assumption 3.1, there exists a stable admissible policy if and only if*

$$\lambda < \frac{1}{1+\frac{1}{\mu}}. \tag{3.1}$$

The sufficiency of this condition is established using a very simple policy: each time a new job arrives, k_n replicas are created and dispatched uniformly at random among the n servers. Under this policy, it is easily checked that each queue behaves as a M/G/1 queue with arrival rate λ and expected service time equal to $1+1/\mu$, which is known to be stable as long as Equation (3.1) is satisfied.

On the other hand, the necessity of Equation (3.1) is established by showing that increasing the number of replicas created cannot enlarge the stability region. The proof of necessity is given in Appendix B.

REMARK 3.1. Note that Equation (3.1) is equivalent to

$$\frac{1}{\lambda} - \frac{1}{\mu} > 1.$$

Thus, the existence of stable admissible policies only depends on the quantity

$$r^* = \frac{1}{\lambda} - \frac{1}{\mu}. \tag{3.2}$$

This is a key quantity that will be ubiquitous throughout this paper.

The condition in Equation (3.1) does not depend on n or k_n, but it critically depends on the assumption that the slowdowns are exponential and independent from the task size. Note the above described policy, which creates only k_n replicas, has the largest stability region (in the sense that the policy is stable for the largest possible set of values of λ). In contrast, as shown in [24], the largest stability region can sometimes be augmented by using more than k_n replicas, when the exponential slowdown assumption is relaxed.

We now introduce a lower bound for the expected delay of a job, under any stable admissible policy.

THEOREM 3.2. *Fix some n. Consider a stable admissible policy, i.e., a policy for which the process $\mathcal{Q}_n(\cdot)$ satisfies assumptions 2.2 and 2.3, and*

under which the process $\mathcal{Q}_n(\cdot)$ admits a unique invariant probability measure π_n. Then, under Assumption 3.1, we have

$$\mathbb{E}_{\pi_n}[W_n] \geq 1 + \frac{1}{\mu} \sum_{i=1}^{k_n} \frac{1}{k_n \left(\frac{1}{\lambda} - \frac{1}{\mu}\right) - i + 1}. \tag{3.3}$$

The proof is given in Appendix C.

A special case of interest is when $k_n \to \infty$ as $n \to \infty$, in which case we have the following result.

COROLLARY 3.3. *Under the same assumptions as in Theorem 3.2, and when $k_n \to \infty$ as $n \to \infty$, we have*

$$\liminf_{n\to\infty} \mathbb{E}_{\pi_n}[W_n] \geq 1 + \frac{1}{\mu} \log\left(\frac{\frac{1}{\lambda} - \frac{1}{\mu}}{\frac{1}{\lambda} - \frac{1}{\mu} - 1}\right). \tag{3.4}$$

Surprisingly, the lower bounds in equations (3.3) and (3.4) are independent from the task size distribution and from the number of servers n. Moreover, it can be shown to hold even when the arrivals form a renewal process. However, it does depend on the number of tasks per job k_n, and on the quantity

$$r^* = \frac{1}{\lambda} - \frac{1}{\mu},$$

which also appeared in Theorem 3.1. Using this notation, we can rewrite the lower bound in Corollary 3.3 as

$$\liminf_{n\to\infty} \mathbb{E}_{\pi_n}[W_n] \geq 1 + \frac{1}{\mu} \log\left(\frac{r^*}{r^* - 1}\right),$$

and the one in Theorem 3.2 as

$$\mathbb{E}_{\pi_n}[W_n] \geq 1 + \frac{1}{\mu} \sum_{i=1}^{k_n} \frac{1}{k_n r^* - i + 1}.$$

Note that, if $k_n r^*$ is an integer, then

$$\frac{1}{\mu} \sum_{i=1}^{k_n} \frac{1}{k_n r^* - i + 1}$$

is the expectation of the k_n-th order statistic of $k_n r^*$ i.i.d. exponential random variables with mean $1/\mu$. Thus, if an admissible policy creates $k_n r^*$

replicas per job, and they all start their service at the same time, the expected service time of the jobs would match the lower bound on the expected delay given in Theorem 3.2. However, the expected delay of a typical job under such a policy might be higher due to queueing delays. In fact, it can be checked that such a policy would result in a system that is critically loaded (i.e., it would have a load of $\rho = 1$ using typical queueing theory notation).

The above observations seem to indicate that no admissible policy could match the lower bound for any finite n. However, even if an admissible policy is not optimal for any finite n, it may be asymptotically optimal as $n \to \infty$. Such policies could be obtained by following the following design principles.

1. Replicas are sent to idle servers whenever possible.
2. For each job, the number of replicas associated with it (r_n) is such that $\frac{r_n}{k_n} \uparrow r^*$ as $n \to \infty$. This forces a heavy-traffic regime through increased replication.
3. The forced heavy-traffic is light enough to ensure that new replicas can be sent to idle servers with high probability.

In the following subsection, we formally introduce policies designed according to these principles.

3.2. *Asymptotically optimal policies.* In this subsection, we introduce a pair of admissible policies that are asymptotically optimal under a mild technical condition on the moment of the task sizes, and on the integrality of r^*k_n. While both policies are stated and can be implemented for any value of k_n, one of them appears to only be tractable for the case of $k_n = 1$.

Intuition behind the policies. While the reasoning at the end of the previous subsection assumed that r^*k_n is an integer, this is not always the case. However, we can always create a random number of replicas for each job so that we create r^*k_n replicas per job in average. In particular, let $p_{k_n} \in (0, 1]$ be such that

$$p_{k_n}\lceil r^*k_n\rceil + (1 - p_{k_n})(\lceil r^*k_n\rceil - 1) = r^*k_n.$$

Then, creating $\lceil r^*k_n\rceil$ replicas per job with probability p_{k_n}, and $\lceil r^*k_n\rceil - 1$ replicas per job with probability $1 - p_{k_n}$, yields an average of r^*k_n replicas per job.

It can be checked that if all replicas associated with the same job start their service at the same time, a policy that creates the number of replicas described above would require all n servers to process the incoming jobs. As a result, such policies would suffer from the same instability problem as

the one mentioned in the previous subsection for the case where r^*k_n is an integer: both subsystems would be critically loaded (i.e., they would have a load of $\rho = 1$ using typical queueing theory notation).

In order to obtain stable policies that have the same asymptotic performance as the ones described above, we force the system to be in heavy-traffic by slightly reducing the average number of replicas created per job (while still creating the same number of replicas per job in the limit as $n \to \infty$), as follows.

Let $\alpha \in (1/2, 1)$ be a constant. Suppose that $\lceil r^*k_n \rceil$ replicas per job are created with probability $\left(p_{k_n} - 2n^{\alpha-1}\right)^+$, and $\lceil r^*k_n \rceil - 1$ replicas per job are created with probability $1 - \left(p_{k_n} - 2n^{\alpha-1}\right)^+$. For all n large enough, this yields an average of $r^*k_n - 2n^{\alpha-1}$ replicas per job, which is slightly less than the r^*k_n replicas per job that would critically load the system. It can be checked that if all replicas associated with the same job start their service at the same time, a policy that creates the number of replicas described above would need

$$\frac{\lambda n}{k_n}\left(p_{k_n} - 2n^{\alpha-1}\right)^+ \left(\lceil r^*k_n \rceil + \frac{k_n}{\mu}\right) \tag{3.5}$$

servers to handle the jobs that start with $\lceil r^*k_n \rceil$ associated replicas, and

$$\frac{\lambda n}{k_n}\left[1 - \left(p_{k_n} - 2n^{\alpha-1}\right)^+\right] \left(\lceil r^*k_n \rceil - 1 + \frac{k_n}{\mu}\right) \tag{3.6}$$

servers to handle the jobs that start with $\lceil r^*k_n \rceil - 1$ associated replicas. Combining these two quantities with the definitions of p_{k_n} and r^*, it leads to a total of

$$n - \frac{2\lambda n^\alpha}{k_n}$$

servers. Thus, there are

$$\frac{2\lambda n^\alpha}{k_n} \tag{3.7}$$

"spare" servers in the system.

Description of the policies. We now introduce the two policies. In order to simplify the analysis, for both policies we partition the n-server system into two subsystems, one with

$$n^{(1)} \triangleq \left\lfloor \frac{\lambda n}{k_n}\left(p_{k_n} - 2n^{\alpha-1}\right)^+ \left(\lceil r^*k_n \rceil + \frac{k_n}{\mu}\right) + \frac{\lambda n^\alpha}{k_n} \right\rfloor$$

servers, and the other one with

$$n^{(2)} \triangleq \left\lceil \frac{\lambda n}{k_n} \left[1 - \left(p_{k_n} - 2n^{\alpha-1}\right)^+\right] \left(\lceil r^* k_n \rceil - 1 + \frac{k_n}{\mu} \right) + \frac{\lambda n^\alpha}{k_n} \right\rceil$$

servers. Note that the number of servers in the first (respectively, second) subsystem is equal to the number of servers required to process the jobs that start with $\lceil r^* k_n \rceil$ (respectively, $\lceil r^* k_n \rceil - 1$) replicas associated with it, given in Equation (3.5) (respectively, Equation (3.6)), plus half of the "spare" servers given in Equation (3.7). Without loss of generality, we assume that $n^{(1)}$ is a multiple of $\lceil r^* \rceil$, and that $n^{(2)}$ is a multiple of $\lceil r^* \rceil - 1$. If this is not the case, we can always choose not to use some of the original n servers. When a job arrives to the system, it is sent to the first subsystem with probability $\left(p_{k_n} - 2n^{\alpha-1}\right)^+$, and to the second subsystem with probability $1 - \left(p_{k_n} - 2n^{\alpha-1}\right)^+$. Then, we do one of the following.

- **Full Replication with Early Cancellation (FREC)**: If at the time of an arrival to the first (second) subsystem there are at least $\lceil r^* k_n \rceil$ (respectively, $\lceil r^* k_n \rceil - 1$) idle servers, then $\lceil r^* k_n \rceil$ (respectively, $\lceil r^* k_n \rceil - 1$) replicas are created and dispatched to idle servers. Otherwise, replicas are created and dispatched to *all* servers in the subsystem. When $\lceil r^* k_n \rceil$ (respectively, $\lceil r^* k_n \rceil - 1$) of these replicas have started their service, all other replicas associated with the same job are cancelled and immediately leave the system. (Note that this is compatible with Assumption 2.3, as canceled replicas have not started their service.)
- **Dummy Queues (DQ)**: If at the time of an arrival to the first (second) subsystem there are at least $\lceil r^* k_n \rceil$ (respectively, $\lceil r^* k_n \rceil - 1$) idle servers among the first $n^{(1)} - \lambda n^\alpha / 2k_n$ (respectively, $n^{(2)} - \lambda n^\alpha / 2k_n$) servers in the subsystem, then $\lceil r^* k_n \rceil$ (respectively, $\lceil r^* k_n \rceil - 1$) replicas are dispatched to those idle servers. Otherwise, k_n replicas are dispatched uniformly at random among the last $\lambda n^\alpha / 2k_n$ servers of the first (respectively, second) subsystem.

Note that when there are enough idle servers, both of these policies send all replicas to idle servers. Thus, if the systems are in light enough heavy-traffic, the vast majority of replicas will be sent to idle servers in both cases. We also note that although the FREC policy seems superior, it appears to be tractable only for the case of $k_n = 1$.

Main results. We now present the main results of this subsection.

THEOREM 3.4. *Suppose that $k_n = 1$, for all n. If*

$$\lambda < \frac{1}{1 + \frac{1}{\mu}},$$

then the FREC policy is stable, and

$$\lim_{n\to\infty} \mathbb{E}[W_n^s] = 1 + \frac{p_1}{\mu\lceil r^* \rceil} + \frac{1 - p_1}{\mu(\lceil r^* \rceil - 1)}.$$

Furthermore, if $\mathbb{E}\big[X^{2+\epsilon}\big] < \infty$ for some $\epsilon > 0$, then

$$\lim_{n\to\infty} \mathbb{E}[W_n^q] = 0.$$

The proof is given in Appendix D.

THEOREM 3.5. *If*

$$\lambda < \frac{1}{1 + \frac{1}{\mu}},$$

then the DQ policy is stable for all n large enough, and:

(i) If $k_n = k$ for all n, then

$$\lim_{n\to\infty} \mathbb{E}[W_n^s] = 1 + \frac{1}{\mu} \sum_{i=1}^{k} \left(\frac{p_k}{\lceil r^* k \rceil - i + 1} + \frac{1 - p_k}{\lceil r^* k \rceil - i} \right). \tag{3.8}$$

Furthermore, if $\mathbb{E}\big[X^{2+\epsilon}\big] < \infty$ for some $\epsilon > 0$, then

$$\lim_{n\to\infty} \mathbb{P}(W_n^q > 0) = 0.$$

(ii) If $k_n \to \infty$ as $n \to \infty$, then

$$\lim_{n\to\infty} \mathbb{E}[W_n^s] = 1 + \frac{1}{\mu} \log\left(\frac{r^*}{r^* - 1} \right). \tag{3.9}$$

Furthermore, if $\mathbb{E}\big[X^{2+\epsilon}\big] < \infty$ for some $\epsilon > 0$, $k_n \in O\big(n^\beta\big)$ for some $\beta < 1/5$, and $\alpha > \max\big\{(2 + 4\beta)/3,\ (1 + 5\beta)/2\big\}$, then

$$\lim_{n\to\infty} \mathbb{P}(W_n^q > 0) = 0.$$

The proof is given in Appendix E.

REMARK 3.2. Recall that if $k_n = k$ for all n, then Theorem 3.2 gives us the lower bound

$$\mathbb{E}[W_n^s] \geq 1 + \frac{1}{\mu} \sum_{i=1}^{k} \frac{1}{r^* k - i + 1}. \tag{3.10}$$

In general, there is a small gap between this lower bound and the asymptotic service time of the DQ policy (Equation (3.8)). However, when r^*k is an integer, there is no gap. This is depicted in Figure 3.2, where we plot the expressions in equations (3.10) and (3.8) as functions of $r^* = 1/\lambda - 1/\mu$ when we keep μ and k fixed and we vary λ.

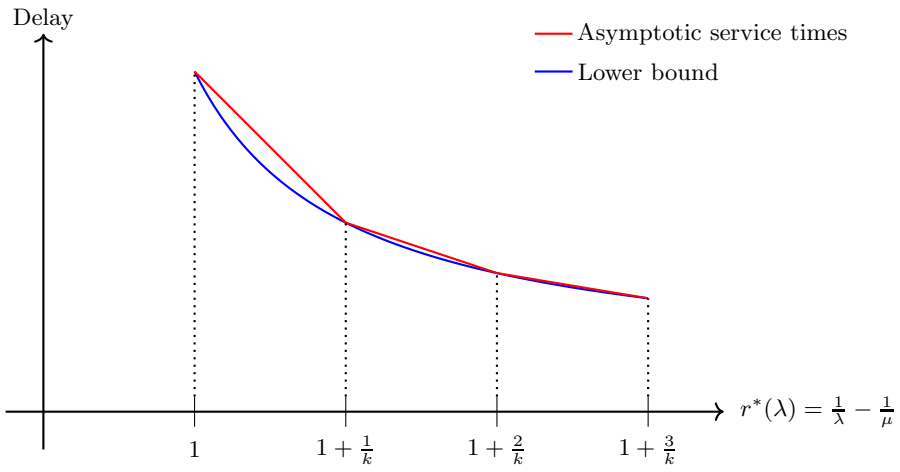

FIG 1. *The asymptotic service time of the DQ policy vs. the corresponding lower bound, as functions of* $r^*(\lambda)$.

On the other hand, if $k_n \to \infty$ as $n \to \infty$, it is easily shown that the lower bound, as $n \to \infty$, is

$$\lim_{n\to\infty} \left[1 + \frac{1}{\mu} \sum_{i=1}^{k_n} \frac{1}{r^* k_n - i + 1} \right] = 1 + \frac{1}{\mu} \log \left(\frac{r^*}{r^* - 1} \right),$$

which coincides with the asymptotic service time of the DQ policy given in Equation (3.9).

REMARK 3.3. In both theorems, the queueing delay W_n^q converges to zero as $n \to \infty$. However, while in Theorem 3.4 it converges to zero in

expectation, in Theorem 3.5 it converges to zero in probability. This weaker result is an artifact of our analysis, and we conjecture that convergence in expectation holds in all cases.

REMARK 3.4. We conjecture that even simpler policies such as Join-Idle-Queue [21], are also asymptotically optimal. However, Join-Idle-Queue is surprisingly hard to analyze in this setting.

3.3. *Main takeaways.* In this section, we obtained a universal lower bound on the expected delay of a typical job, and designed and analyzed dispatching/replication policies that match the universal lower bound, at least in the limit as $n \to \infty$, as long as $r^* k_n$ either is an integer, or diverges as $n \to \infty$. This establishes both the asymptotic optimality of our policies, as well as the tightness of our lower bound, in those cases. It is unclear whether our policies are always asymptotically optimal. However, there appears to be some slack in certain inequalities in the proof of the delay lower bound (Theorem 3.2), which suggests that our policies may indeed be always asymptotically optimal.

It is worth noting that the average number of replicas created per task under our policies is always equal to $r^* = 1/\lambda - 1/\mu$. In particular, this is independent of the task size distribution, of the realization of the task sizes, and of the current state of the queues. Combined with relatively simple dispatching rules for the replicas, this makes the aforementioned policies simple enough to be practical.

Finally, note that the lower bounds obtained in Theorem 3.2 and Corollary 3.3 are increasing in the arrival rate λ. Since the lower bounds ignore possible queueing delays, the fact that they are increasing in λ is solely due to the fact that higher arrival rates allow less overall replication (i.e., higher arrival rates lead to smaller r^*). Thus, this arrival rate versus delay performance tradeoff is a fundamental limitation of this kind of systems, and not a reflection of the usual effect of congestion on delays.

4. General size-based slowdowns. In this section, we introduce a new model for the slowdowns, and obtain asymptotically optimal policies within a somewhat restricted set of admissible policies.

For certain applications, the assumption that the slowdown is independent from the task size is not realistic. Indeed, if replicas are very large, one can argue that the server side variability is "averaged out", which then results in less variable service times. By the same argument, if replicas are very small, it is more likely for them to complete their service within a time period when

the server is atypically slow or fast, and service times become more variable. This intuition is somewhat validated by the fact that the higher variability of smaller tasks is widely observed in cloud computing systems operated by Facebook and Microsoft [2, 1].

In order to model these size-based slowdowns, we introduce an appropriate dependence structure between the slowdowns and the replicas' sizes.

ASSUMPTION 4.1. For $r \geq 1$ and for $i \leq r$, let $S_{[i:r]}$ be the i-th order statistic of slowdowns $S_1, \ldots, S_r$ corresponding to a single job with replicas of size X. We assume the following.

1. For all $r \geq 1$, and all $x \geq 0$, we have
$$\mathbb{E}\big[S_r \,\big|\, X = x\big] = \frac{1}{\mu}.$$
2. For all $r \geq 2$ and $i \leq r$, the expression
$$\mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] - \frac{1}{\mu}$$
is either increasing or decreasing in x. Furthermore,
$$\lim_{x\to\infty} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] = \frac{1}{\mu},$$
for all $r \geq 2$ and $i \leq r$.
3. For all $r \geq 1$ and $i \leq r$, the expression
$$\mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] - \mathbb{E}\big[S_{[i:r+1]} \,\big|\, X = x\big]$$
is decreasing in x.

Part 1 states that the expectation of the slowdowns does not depend on the task sizes. Part 2 reflects the premise that bigger tasks observe less variable slowdowns, and thus their order statistics converge monotonically to their mean. For instance, we might be dealing with a situation where the variance of the slowdowns converges to 0 as $x \to \infty$. Finally, Part 3 implies that the improvement brought by an extra replica is smaller for larger tasks.

EXAMPLE 4.1. Suppose that the slowdowns have a Gamma distribution with shape parameter x, and rate parameter μ/x. It is straightforward (but tedious) to check that these slowdowns satisfy Assumption 4.1.

4.1. *Block policies.* In order to keep the problem tractable, we restrict ourselves to a subset of the admissible policies introduced in Subsection 2.1, those that satisfy the following assumption.

ASSUMPTION 4.2 (Block policy). All the replicas associated with the same job start their service at the same time.

For the case $k_n = 1$, this assumption can be satisfied by sending replicas to different queues in a careful way so that it is guaranteed that they will start their service at the same time. An example of this is the FREC policy introduced in Subsection 3.2, which is shown to be a Block policy in Lemma D.1. On the other hand, Assumption 4.2 excludes policies where the start times of the replicas are staggered in time, which could yield better performance in some cases.

Unfortunately, analyzing the class of general admissible policies appears to be difficult. For this reason, we restrict ourselves to Block policies, which are easier to analyze.

4.2. *Delay lower bound.* In this subsection, we present a straightforward lower bound on the expected delay of a typical job under Block policies, as the solution of a minimization problem. In order to do this, we consider a relaxation of the problem in which we have an infinite number of servers, but where there can only be n busy servers in expectation, in steady-state.

First, note that the delay of a typical job is trivially lower bounded by the service time of a typical job. Since all replicas associated with the same job start their service at the same time (Assumption 4.2), and since no replica with elapsed time can be prematurely cancelled (Assumption 2.3), the service time of a typical job under a Block policy is completely determined by the (possibly random) number of replicas that start their service. In particular, the service time of a typical job is determined by a (policy specific) measurable function $p : \mathbb{R}_+ \to [0,1]^\infty$, where $p_r(x)$ is the probability that a typical job with tasks of size x has r replicas associated with it that get created and start their service. Using this, we obtain a lower bound on the expected delay of a typical job under a Block policy, as the solution of an optimization problem.

LEMMA 4.1. *Fix some n, and consider an admissible policy (i.e., a policy for which the process $\mathcal{Q}_n(\cdot)$ satisfies assumptions 2.2 and 2.3), that satisfies Assumption 4.2, and under which the process $\mathcal{Q}_n(\cdot)$ admits at least one*

invariant probability measure π_n. Then, under Assumption 4.1, we have

$$\mathbb{E}_{\pi_n}[W_n] \geq \inf_{p \in \mathcal{P}_{k_n}} \int_0^\infty x \sum_{r=k_n}^\infty p_r(x)\Big(1 + \mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big]\Big) d\mathbb{P}_X(x) \tag{4.1}$$

$$s.t. \quad \frac{\lambda}{k_n} \int_0^\infty x \sum_{r=k_n}^\infty p_r(x)\Big[r + (r-k_n)\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] + \sum_{i=1}^{k_n} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big]\Big] d\mathbb{P}_X(x) \leq 1, \tag{4.2}$$

where $\mathbb{P}_X$ is the distribution of the task sizes, and $\mathcal{P}_{k_n}$ is the set of measurable functions from $\mathbb{R}_+$ to the infinite-dimensional simplex

$$P_{k_n} \triangleq \left\{ p \in [0,1]^\infty : \sum_{r=k_n}^\infty p_r = 1 \right\}.$$

The proof is given in Appendix F.

The optimization problem defined by equations (4.1) and (4.2) corresponds to the minimization of the service time of a typical job over all possible distributions p for the number of replicas that are created and start service, subject to the constraint that the average number of busy servers must be at most n.

In general, there is no closed form solution for the optimization problem in Lemma 4.1. However, in the following subsection we will derive some properties of optimal solutions in order to design delay optimal policies and understand their behavior.

4.3. *Properties of optimal solutions.* If p^* is a feasible solution of the optimization problem defined by equations (4.1) and (4.2) that attains the minimum, we are interested in characterizing how it depends on x, and k_n.

To begin with, depending on the parameters of the system, the optimization problem defined by equations (4.1) and (4.2) may not have feasible solutions. A sufficient condition for the existence of feasible solutions is established in the following lemma.

LEMMA 4.2. *If*

$$\lambda \leq \frac{1}{1 + \frac{1}{\mu}}, \tag{4.3}$$

then the optimization problem defined by equations (4.1) *and* (4.2) *has a finite minimum, which is attained.*

The proof is given in Appendix G.

Note that this sufficient condition for feasibility does not depend on n, k_n, nor on the task size distribution.

REMARK 4.1. Surprisingly, the condition of Equation (4.3) for the existence of a feasible solution is the same as the one in Theorem 3.1 for the existence of a stable policy, which concerned the model with independent exponential slowdowns. However, unlike Theorem 3.1, this condition may not be necessary under this model.

REMARK 4.2. Note that any feasible point $\tilde{p}$ such that $\tilde{p}(x) = p^*(x)$ for almost every x (with respect to $\mathbb{P}_X$) is also optimal. Therefore, any necessary properties of the optimal solutions are only true almost everywhere with respect to $\mathbb{P}_X$.

The following result states some properties of the optimal solutions for arbitrary values of k_n.

THEOREM 4.3. *Every feasible point p that achieves the minimum in the problem defined by equations* (4.1) *and* (4.2) *is equal almost everywhere (with respect to $\mathbb{P}_X$) to a point p^* that satisfies the following:*

(i) $p^*_{k_n}(x) = 1$ *for all x large enough.*
(ii) There exists $\overline{r}_{k_n} \in O(k_n)$ *such that* $p^*_r(x) = 0$ *for all* $r \geq \overline{r}_{k_n}$ *and* $x \geq 0$.

The proof is given in Appendix H.

REMARK 4.3. This result shows us that smaller tasks should be replicated more (although only up to a constant number of replicas per task), and that very large tasks should not be replicated at all. This conclusion is consistent with and provides support for current practice [2, 1]. Note however that this conclusion can only be reached when the slowdowns depend on the task size; it does not hold under independent slowdowns, as in the $S\&X$ model [11].

4.3.1. *Case $k_n = 1$.* For the case where $k_n = 1$ for all n, we can refine the results in Theorem 4.3 under the following additional assumption on the distribution of the slowdowns.

ASSUMPTION 4.3. The expression

$$r\mathbb{E}\left[S_{[1:r]} \,\middle|\, X = x\right]$$

is convex as a function of the integer parameter r, for all $x \geq 0$.

The following lemma provides a sufficient condition for this assumption to hold.

LEMMA 4.4. *If the expression*

$$\mathbb{P}\left(S > \frac{s}{r} \,\middle|\, X = x\right)^r \tag{4.4}$$

is convex as a function of the integer parameter r, for all $s, x > 0$, then Assumption 4.3 holds.

For example, the exponential and the Pareto distributions satisfy Equation (4.4). In general, distributions with tails heavier than the exponential will also satisfy it, and therefore will also satisfy Assumption 4.3.

THEOREM 4.5. *Suppose that $k_n = 1$ for all n, and that the distributions of the slowdowns satisfy Assumption 4.3. Then, every feasible point p that achieves the minimum in the problem defined by equations (4.1) and (4.2) is equal almost everywhere (with respect to $\mathbb{P}_X$) to a point p^* that satisfies the following:*

(i) For every x, $p^*(x)$ is concentrated on up to two consecutive integers.
(ii) If $\mathbb{P}_X$ is non-atomic, then $p^*(x)$ is concentrated in a single integer, for all x.
(iii) The expected number of replicas,

$$r^*(x) \triangleq \sum_{r=1}^{\infty} r p_r^*(x),$$

is nonincreasing with x.

The proof is given in Appendix I.

REMARK 4.4. For the special case of $k_n = 1$, Theorem 4.5 provides a crisper characterization of the optimal solutions p^* than Theorem 4.3. In particular, it states that the optimal number of replicas is a nonincreasing function of the task size.

In the next subsection, we will use the properties derived in this subsection to design an asymptotically optimal Block policy.

4.4. *Asymptotically optimal Block policies.* In this subsection, we show that the lower bound on the delay obtained in the previous subsection is asymptotically attainable using an appropriate sequence of Block policies based on the FREC and DQ policies introduced in Subsection 3.2.

Suppose that

$$\lambda < \frac{1}{1+\frac{1}{\mu}}.$$

Let $\alpha \in (1/2, 1)$ be a constant, and let $p^{(n)}$ be a solution of the following optimization problem:

$$\begin{aligned}
\inf_{p\in\mathcal{P}_{k_n}} \quad & \int_0^\infty x \sum_{r=k_n}^\infty p_r(x)\Big(1+\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X=x\big]\Big)d\mathbb{P}_X(x) \\
s.t. \quad & \frac{\lambda}{k_n}\int_0^\infty x \sum_{r=k_n}^\infty p_r(x)\left[r+(r-k_n)\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X=x\big]\right. \\
& \qquad \left. +\sum_{i=1}^{k_n}\mathbb{E}\big[S_{[i:r]} \,\big|\, X=x\big]\right] d\mathbb{P}_X(x) \le 1-n^{\alpha-1}.
\end{aligned}$$

Note that, by moving the $1-n^{\alpha-1}$ to the other side, we get the same optimization problem as in Lemma 4.1, with λ replaced by $\lambda/(1-n^{\alpha-1})$. Furthermore, for all n large enough, we have

$$\frac{\lambda}{1-n^{\alpha-1}} \le \frac{1}{1+\frac{1}{\mu}}.$$

Thus, Lemma 4.2 applies, and a solution $p^{(n)}$ is guaranteed to exist. Moreover, recall that Theorem 4.3 states that there exists $\overline{r}_{k_n}^{(n)} \in O(k_n)$ such that $p_r^{(n)}(x) = 0$ for all $r \ge \overline{r}_{k_n}^{(n)}$ and $x \ge 0$. In other words, $p^{(n)}$ is concentrated in its first $\overline{r}_{k_n}^{(n)} - k_n + 1$ indices.

Let us partition the n-server system into $\overline{r}^{(n)}_{k_n} - k_n + 1$ subsystems, indexed by $r = k_n, \ldots, \overline{r}^{(n)}_{k_n}$, such that the r-th subsystem has

$$n^{(r)} \triangleq \left\lfloor \frac{\lambda n}{k_n} \int_0^\infty x p_r^{(n)}(x) \left[r + (r-k_n)\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] + \sum_{i=1}^{k_n} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] \right] d\mathbb{P}_X(x) + \frac{n^\alpha}{\overline{r}^{(n)}_{k_n}} \right\rfloor$$

servers. As it was the case for the policies introduced in Subsection 3.2, the number of servers in the r-th subsystem is equal to the number of servers required to handle the incoming jobs to the subsystem, namely,

$$\frac{\lambda n}{k_n} \int_0^\infty x p_r^{(n)}(x) \left[r + (r-k_n)\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] + \sum_{i=1}^{k_n} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] \right] d\mathbb{P}_X(x),$$

plus a fraction of the "spare" servers,

$$\frac{n^\alpha}{\overline{r}^{(n)}_{k_n}}.$$

Without loss of generality, we assume that $n^{(r)}$ is a multiple of r, for all r. If this is not the case, we can always choose not to use some of the original n servers.

We now introduce two policies. In both, when a job with tasks of size x arrives to the system, it is sent to the r-th subsystem with probability $p_r^{(n)}(x)$. Then, we do one of the following.

- **Size-Based Full Replication with Early Cancellation (SB-FREC)**: If at the time of an arrival to the r-th subsystem there are at least r idle servers, then r replicas are created and dispatched to idle servers. Otherwise, replicas are created and dispatched to *all* servers in the subsystem. When r of these replicas have started their service, all other replicas associated with the same job are cancelled and immediately leave the system.

- **Size Based Dummy Queues (SB-DQ)**: If at the time of an arrival to the r-th subsystem there are at least r idle servers among the first $n^{(r)} - n^{\alpha}/2\overline{r}_{k_n}^{(n)}$ servers in the subsystem, then r replicas are dispatched to those idle servers. Otherwise, if there are at least r idle servers among the last $n^{\alpha}/2\overline{r}_{k_n}^{(n)}$ servers of the r-th subsystem, then r replicas are dispatched to those idle servers. Otherwise, jobs are queued in a virtual FIFO queue at the dispatcher until there are enough idle servers among the last $n^{\alpha}/2\overline{r}_{k_n}^{(n)}$ servers of the r-th subsystem.

Note that when there are enough idle servers, both of these policies send all replicas to idle servers. Thus, if the systems are in light enough heavy-traffic, the vast majority of replicas will be sent to idle servers in both cases. We also note that although the FREC policy seems superior, it appears to be tractable only for the case of $k_n = 1$.

We now present the main results of this subsection.

THEOREM 4.6. *Suppose that $k_n = 1$ for all n, and that*

$$\lambda < \frac{1}{1 + \frac{1}{\mu}}.$$

Then, the SB-FREC policy is stable for all n large enough, and we have

$$\lim_{n\to\infty} \mathbb{E}\big[W_n^s\big] = \int_0^\infty x \sum_{r=1}^{\infty} p_r^*(x)\Big(1 + \mathbb{E}\big[S_{[1:r]} \,\big|\, X = x\big]\Big) d\mathbb{P}_X(x),$$

where p^* is a solution of the optimization problem defined by equations (4.1) and (4.2), for $k_n = 1$. Furthermore, if

$$\mathbb{E}\left[\Big(XS\Big)^{2+\epsilon}\right] < \infty$$

for some $\epsilon > 0$, we also have

$$\lim_{n\to\infty} \mathbb{E}\big[W_n^q\big] = 0.$$

The proof is given in Appendix J.

THEOREM 4.7. *Suppose that $k_n = k$ for all n, and that*

$$\lambda < \frac{1}{1 + \frac{1}{\mu}}.$$

Then, the SB-DQ policy is stable for all n large enough, and we have

$$\lim_{n\to\infty} \mathbb{E}\big[W_n^s\big] = \int_0^\infty x \sum_{r=k}^{\infty} p_r^*(x)\Big(1 + \mathbb{E}\big[S_{[k:r]} \,\big|\, X = x\big]\Big) d\mathbb{P}_X(x),$$

where p^* is a solution of the optimization problem defined by equations (4.1) and (4.2), for $k_n = k$. Furthermore, if

$$\mathbb{E}\left[\Big(XS_{[k:k]}\Big)^{2+\epsilon}\right] < \infty$$

for some $\epsilon > 0$, we also have

$$\lim_{n\to\infty} \mathbb{P}\big(W_n^q > 0\big) = 0.$$

The proof is given in Appendix J.

Remark 4.5. Theorems 4.6 and 4.7 imply that the lower bound for the expected delay of a typical job established in Theorem 4.1 is asymptotically attained. Moreover, since the asymptotically delay optimal polices were constructed by using the solutions to a suitable version of the optimization problem that serves as the lower bound, the properties derived in theorems 4.3 and 4.5 (such as the fact that smaller tasks should be replicated more than large tasks) still hold and are therefore properties of our asymptotically delay-optimal policies.

4.5. *Main takeaways.* In this section, we obtained a lower bound on the expected delay of a typical job under a restricted class of policies, as the solution of a certain optimization problem. Moreover, we analyzed the properties of its optimal solutions to guide the design of asymptotically optimal dispatching/replication policies.

Similar to the case of independent exponential slowdowns studied in Section 3.1, the number of replicas created per task is independent of the task size distribution, and of the current state of the queues. However, the number of replicas created for each incoming job now depends on the realization of the task sizes. This is a consequence of the dependence of the slowdowns on the size of the tasks (Assumption 4.1), which was not present in simpler models.

Finally, note that the constraint of the optimization problem that describes the asymptotic delay performance of our policies (Equation (4.2)) depends on the arrival rate λ in a way that a larger λ can only increase

the delay of the policy. Since the asymptotic delays of our policies are just their asymptotic service times, the fact that they are increasing in λ is solely due to the fact that higher arrival rates allow less overall replication (i.e., higher arrival rates lead to smaller r^*). This was also observed in Section 3 under the independent exponential slowdowns mode, which suggests that this arrival rate versus delay performance tradeoff is a fundamental limitation of replication-based systems, which transcends the model used for the distribution of the slowdowns.

5. Conclusions and future work. The main objective of this paper was to study the impact of replication on the delay performance of distributed service systems with random server slowdowns. We did this under two different models for the server slowdowns, inspired by different applications.

When the server slowdowns are independent from the task sizes, we showed that the asymptotic expected delay is minimized when the number of replicas created per task is equal to a constant that only depends on the arrival rate, and on the expected slowdown. Surprisingly, it is independent from the number of tasks per job, and from the inter-arrival and task size distributions.

On the other hand, when the server slowdowns depend on the task sizes in some particular way, we showed that the asymptotic expected delay is minimized (among all policies where replicas associated with the same job start their service at the same time) when smaller tasks are replicated more than larger tasks.

There are several interesting directions for future research. For example:

(i) In this paper, as well as in most of the literature, all servers have the same average processing rate. Thus, one interesting future line of work would be to explore the impact of heterogeneity in the average processing rates of servers in replication systems.

(ii) On a similar note, another interesting possible research direction would be to study the impact of heterogeneous jobs, where not all jobs can be served by the same number of servers (or at least not at the same speed).

(iii) We conjecture that even simpler policies such as Join-Idle-Queue [21], are also asymptotically optimal. However, Join-Idle-Queue is surprisingly hard to analyze in this setting.

APPENDIX A: CONVENIENT NOTATION

In this appendix, we introduce some definitions and notation that will be used in the proofs of theorems 3.1 and 3.2.

For $j \geq 1$, we denote the **relative service start times** of the replicas associated with the j-th job by $T_{j,r}$, ordered so that $0 \leq T_{j,1} \leq \cdots \leq T_{j,n+k_n-1} \leq \infty$. More precisely, $T_{j,r}$ is the (random) time, relative to the arrival of the j-th job, when the r-th replica associated with it starts its service. As a convention, we set $T_{j,r}(\omega) = \infty$ if the r-th replica associated with the j-th job does not start its service under the sample path ω. The times $T_{j,r}$ are mostly determined by the policy, but also depend on the queueing delays of replicas, and potentially on the service times of replicas associated with the same job that have already finished their service.

For $j \geq 1$ and for $i = 1, \ldots, k_n$, we denote the **relative departure times** of the replicas associated with the j-th job by $D_j^{(i)}$, ordered so that $0 \leq D_j^{(1)} \leq \cdots \leq D_j^{(k_n)} \leq \infty$. More precisely, $D_j^{(i)}$ is the time, relative to the arrival of the j-th job, when i replicas associated with it finish their service. Since replicas cannot be cancelled after they start service (Assumption 2.3), $D_j^{(i)}$ is the i-th order statistic of the random variables

$$\left\{T_{j,r} + X_j\big(1 + S_{j,r}\big)\right\}_{r=1}^{n+k_n-1}.$$

In particular, $D_j^{(k_n)}$ is the delay of the j-th job. Moreover, we use the convention $D_j^{(0)} = 0$.

The rest of the notation and definitions are only used for the proof of Theorem 3.2.

A.1. Partition of the jobs in service. We now introduce a dynamic partition of the set of jobs that are in service (i.e., the jobs that have at least one replica associated with it in service), by assigning a phase to each job in service. At any given time:

a) A job with tasks of size x is in phase 0 if at least one replica associated with it has started its service, and if all of the replicas associated with it have been in service for less than x units of time.
b) For $i = 1, \ldots, k_n$, a job with tasks of size x is in phase i if it is not in phase 0 (i.e., if at least one replica associated with it, past or present, has been in service at least x units of time) and if exactly $i-1$ replicas associated with it have finished their services.

These phases define a partition of the jobs that are in service, consisting of $k_n + 1$ subsets (or phases). Moreover, since this is a dynamical system, jobs not only start and finish their services, but they also change phases. In particular, all jobs start their services in phase 0, they can only leave the system if they are in phase k_n, and they go through all the intermediate phases in ascending order (possibly spending 0 time in some phases). The possible transitions between the phases are depicted in Figure A.1.

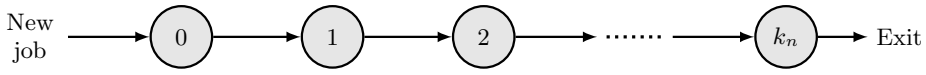

FIG 2. *Possible transitions of jobs in service between the phases.*

Suppose that the j-th job arrives to the system at time 0. Then:

a) The j-th job is in phase 0 between the time that the first replica associated with it starts its service ($T_{j,1}$), and the time that the first replica associated with it has spent X_j units of time in service ($T_{j,1} + X_j$). At that point, the job moves to phase 1.
b) The j-th job is in phase 1 between the time that the first replica associated with it is in service for X_j units of time ($T_{j,1} + X_j$), and the first time that one of the replicas associated with it finishes its service. At that point, it moves to phase 2.
c) For $i = 2, \ldots, k_n$, the j-th job is in phase i between the $(i-1)$-th time that one of the replicas associated with it finishes its service ($D_j^{(i-1)}$), and the i-th time that one of the replicas associated with it finishes its service ($D_j^{(i)}$). At that point, the job either moves to phase $i+1$ (if $i < k_n$) or leaves the system (if $i = k_n$).

A.2. Partition of the replicas in service. We now introduce similar terminology for the replicas that are in service. First, we define two types of replicas.

a) A replica of size x is *wasteful* if is in service less than x units of time in total. In particular, the r-th replica of the j-th job is wasteful if and only if $D_j^{(k_n)} < T_{j,r} + X_j$.
b) A replica of size x is *useful* if it is in service for at least x units of time in total. In particular, the r-th replica of the j-th job is useful if and only if $D_j^{(k_n)} \geq T_{j,r} + X_j$.

Note that the type of a replica is future-dependent, noncausal property. As such, it does not change with the passage of time.

Second, we define a partition of the useful replicas in service by assigning a phase to each one of them, as follows. At any given time:

a) A useful replica of size x is in phase 0 if it has been in service less than x units of time.
b) For $i = 1 \ldots, k_n$, a useful replica of size x is in phase i if it has been in service for at least x units of time, and if it is associated with a job in phase i.

Note that this indeed defines a partition of all the useful replicas in service into k_n+1 subsets (or phases). Moreover, as the system evolves, useful replicas change phases over time. In particular, the transitions between phases are as follows. Suppose that the j-th job arrives to the system at time 0. The r-th replica associated with the j-th job starts its service at time $T_{j,r}$, as long as this time comes before the job leaves the system, i.e., only if $T_{j,r} < D_j^{(k_n)}$. Assuming that this is the case, and that the replica is useful (i.e., that $T_{j,r} + X_j \leq D_j^{(k_n)}$ and thus it spends at least X_j units of time in service), we have the following.

a) The r-th replica associated with the j-th job is in phase 0 between the time that it starts its service ($T_{j,r}$), and the time that it spends X_j units of time in service ($T_{j,r}+X_j$). At that point, it moves to one of the phases $1, \ldots, k_n$. In particular, it moves to phase i if it is associated with a job in phase i.
b) The r-th replica associated with the j-th job is in phase 1 between the time when it has spent X_j units of time in service ($T_{j,r} + X_j$), and the first time that a replica associated with the same job has finished its service ($D_j^{(1)}$). At that point, the replica either leaves the system (if it has finished its service), or moves to phase 2 (if it has not finished its service).
c) For $i = 2, \ldots, k_n$, the r-th replica associated with the j-th job is in phase i between the time when it has spent X_j units of time in service and $i-1$ replicas associated with the same job have already finished service ($\max\{T_{j,r} + X_j, D_j^{(i-1)}\}$), and the i-th time that a replica associated with the same job finishes its service ($D_j^{(i)}$). At that point, the replica either leaves the system (if it has finished its service or if $i = k_n$), or moves to phase $i+1$ (if it has not finished its service and $i < k_n$).

The possible transitions between phases are depicted in Figure A.2.

For $j \geq 1$, and for $i = 1, \ldots, k_n$, let $T_{j,r}^{(i)}$ be the **relative phase start times** of the replicas in phase i associated with the j-th job, arranged so

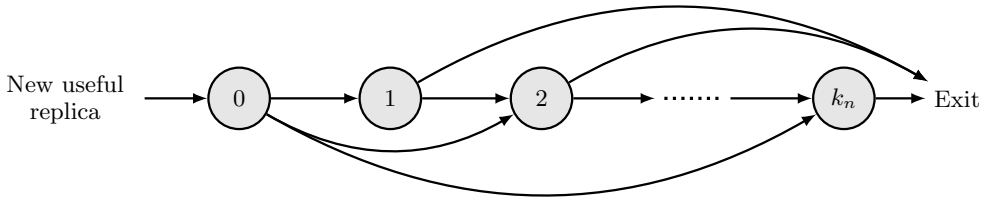

FIG 3. *Possible transitions of replicas in service between the phases.*

that $0 \leq T_{j,1}^{(i)} \leq \cdots \leq T_{j,n}^{(i)} \leq \infty$. More precisely, $T_{j,r}^{(i)}$ is the time elapsed between the moment when the j-th job starts phase i, and the moment when the r-th replica associated with it enters phase i. As a convention, we set $T_{j,r}^{(i)}(\omega) = \infty$ if the r-th replica associated with the j-th job is never in phase i under the sample path ω. These times are mostly determined by the policy, but also depend on the queueing delays of replicas, and potentially on the service times of replicas associated with the same job that have already finished their services.

Furthermore, for $j \geq 1$ and for $i = 1, \ldots, k_n$, let $S_{j,1}^{(i)}, \ldots, S_{j,n}^{(i)}$ be such that $X_j S_{j,1}^{(i)}, \ldots, X_j S_{j,n}^{(i)}$ are the **remaining service times** of the replicas in phase i associated with the j-th job. That is, $X_j S_{j,r}^{(i)}$ is the remaining service time of the r-th replica associated with the j-th job, when the replica starts phase i. Thus, for $i = 1, \ldots, k_n$, the time that the j-th job spends in the i-th phase is equal to

$$\min_{r=1,\ldots,n} \left\{ T_{j,r}^{(i)} + X_j S_{j,r}^{(i)} \right\}.$$

We illustrate the concepts and definitions introduced above with the following example.

EXAMPLE A.1. Suppose that $k_n = 2$. In Figure 4 we depict the evolution of the phases of the j-th job, and of its replicas, for a sample path of a certain policy. In our example, only the first four replicas that start their service are useful, while the fifth one is wasteful. Moreover, the fourth replica is never in phase 1.

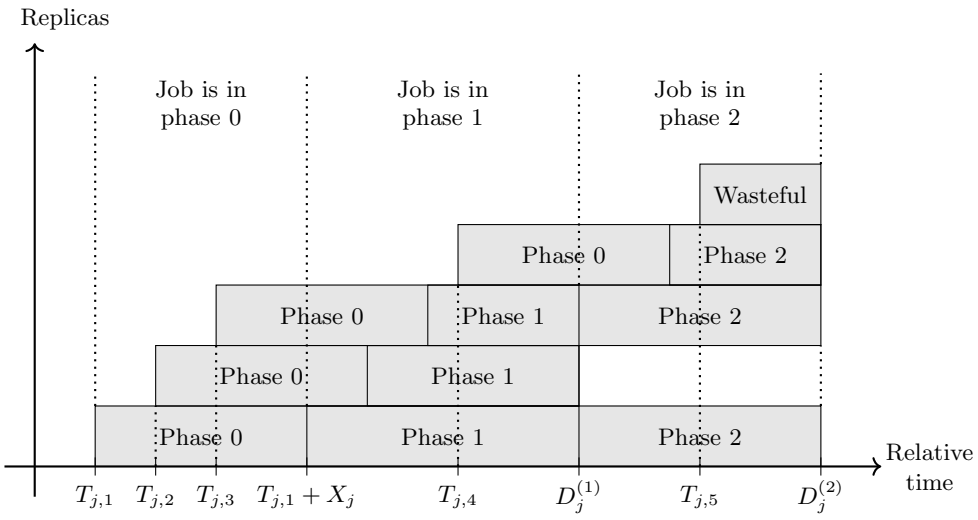

FIG 4. *Evolution of phases for a particular sample path of a given policy.*

In the example, the relative phase start times of replicas, for phase 1, are

$$\begin{aligned} T_{j,1}^{(1)} &= 0 \\ T_{j,2}^{(1)} &= T_{j,2} - T_{j,1} \\ T_{j,3}^{(1)} &= T_{j,3} - T_{j,1}, \end{aligned}$$

and the relative phase start times of replicas, for phase 2, are

$$\begin{aligned} T_{j,1}^{(2)} &= 0 \\ T_{j,2}^{(2)} &= 0 \\ T_{j,3}^{(2)} &= T_{j,4} - D_j^{(1)}. \end{aligned}$$

APPENDIX B: PROOF OF THE NECESSARY CONDITION IN THEOREM 3.1

A necessary condition for the existence of an invariant probability measure for the extended queue state process $\mathcal{Q}_n(\cdot)$ is that the total workload in the queues does not diverge with time. In the parallel queueing model that we consider, the total workload can only remain bounded if the average rate at which workload comes into the queues is strictly less than the system's total processing power (see [4], for example). This means that the arrival

rate of jobs, times the expected workload of a typical job, must be strictly less than the total processing power of the system. Since the arrival rate is $\lambda n/k_n$, and the total processing power is n, the stability condition is

$$\left(\frac{\lambda n}{k_n}\right)\big(\text{Expected workload of typical job}\big) < n. \tag{B.1}$$

Note that the workload of a job in a replication system depends on the policy, as is defined as the sum of all the service times of the replicas associated with it. With this in mind, we now proceed to obtain a lower bound for the expected workload of a typical job under an admissible policy, using the notation introduced in Section A.

Let $0 \le T_1 \le \cdots \le T_{n+k_n-1} \le \infty$ be the steady-state relative service start times of the replicas associated with a typical job with tasks of size X, and slowdowns $S_1, \ldots, S_{n+k_n-1}$. For $i = k_n, \ldots, n+k_n-1$ and for $r = 1, \ldots, k_n$, let $D_{[r:i]}$ be the r-th order statistic of the set of random variables

$$\big\{T_1 + X(1+S_1), \ldots, T_i + X(1+S_i)\big\}.$$

Note that $D_{[1:i]} \le \cdots \le D_{[k_n:i]}$ are the relative departure times of a policy with relative service start times $0 \le \hat{T}_1 \le \cdots \le \hat{T}_{n+k_n-1} \le \infty$, such that $\hat{T}_r = T_r$, for all $r \le i$, and $\hat{T}_r = \infty$, for all $r > i$.

For $i = k_n, \ldots, n+k_n-1$, let

$$M_i \triangleq \sum_{r=1}^{i} \min\Big\{X(1+S_r),\ \big(D_{[k_n:i]} - T_r\big)^+\Big\}.$$

This is the workload of a typical job (i.e., the sum of the service times of all the replicas associated with it) under a policy with the relative service start times $0 \le \hat{T}_1 \le \cdots \le \hat{T}_{n+k_n-1} \le \infty$, such that $\hat{T}_r = T_r$, for all $r \le i$, and $\hat{T}_r = \infty$, for all $r > i$. In particular, M_{n+k_n-1} is the workload of a typical job under the original policy, so we are interested in finding a lower bound for it. With this in mind, consider the decomposition

$$M_{n+k_n-1} = M_{k_n} + \sum_{i=k_n+1}^{n+k_n-1} \Big[M_i - M_{i-1}\Big]. \tag{B.2}$$

The rest of the proof consists of obtaining an expression for the expectation of the first term on the right-hand side, and showing that the expectation of the rest of the terms in the right-hand side is nonnegative.

First, note that $T_r + X(1 + S_r) \leq D_{[k_n:k_n]}$, for all $r \leq k_n$, and thus $X(1 + S_r) \leq \left(D_{[k_n:k_n]} - T_r\right)^+$, for all $r \leq k_n$. It follows that

$$M_{k_n} = \sum_{r=1}^{k_n} X(1 + S_r).$$

Combining this with the fact that $S_1, \ldots, S_{k_n}$ are independent exponential random variables with mean $1/\mu$, independent from X (Assumption 3.1), we get that

$$\mathbb{E}[M_{k_n}] = \mathbb{E}\left[\sum_{r=1}^{k_n} X(1 + S_r)\right] = k_n \left(1 + \frac{1}{\mu}\right). \tag{B.3}$$

On the other hand, note that the difference of workloads $M_i - M_{i-1}$ is how much more the servers have to work when up to i replicas are ever made, compared to when only up to $i - 1$ replicas are ever made. It is not immediately clear whether this quantity is positive or not. If the i-th replica indeed starts its service at some point, it adds workload to its assigned server. However, if the i-th replica finishes its service before k_n of the first $i - 1$ finish theirs, the i-th replica contributes to the premature cancellation of other replicas, effectively reducing the workloads of their servers. We will now show that, in expectation, the difference in workload is nonnegative.

Let us fix i, and consider the following partition of the sample space. Let E_1 be the event that k_n replicas finish their service before the i-th replica spends X units of time in service, i.e., the event that $D_{[k_n:i-1]} \leq T_i + X$. Moreover, for $\ell = 0, \ldots, k_n - 1$, let $E_2(\ell)$ be the event that $\ell < k_n$ replicas finished their service by the time the i-th replica spends X units of time in service, i.e., the event that $D_{[\ell:i-1]} \leq T_i + X < D_{[\ell+1:i-1]}$ (using the convention that $D_{[0:i-1]} = 0$). We show that the expectation of the difference in workloads is nonnegative, for each of the events defined above, in the following two lemmas.

LEMMA B.1. *If* $\mathbb{P}(E_1) > 0$, *then* $\mathbb{E}\big[M_i - M_{i-1} \mid E_1\big] \geq 0$.

PROOF. First note that in the event E_1, we have $D_{[k_n:i-1]} \leq T_i + X(1+S_i)$

and thus $D_{[k_n:i-1]} = D_{[k_n:i]}$. It follows that

$$\begin{aligned}
\mathbb{E}\big[M_i - M_{i-1} \mid E_1\big] &= \mathbb{E}\left[\sum_{r=1}^{i} \min\left\{X(1+S_r),\ \big(D_{[k_n:i]} - T_r\big)^+\right\} \,\middle|\, E_1\right] \\
&\quad - \mathbb{E}\left[\sum_{r=1}^{i-1} \min\left\{X(1+S_r),\ \big(D_{[k_n:i-1]} - T_r\big)^+\right\} \,\middle|\, E_1\right] \\
&= \mathbb{E}\left[\min\left\{X(1+S_i),\ \big(D_{[k_n:i]} - T_i\big)^+\right\} \,\middle|\, E_1\right] \\
&\geq 0.
\end{aligned}$$

□

LEMMA B.2. *If* $\mathbb{P}\big(E_2(\ell)\big) > 0$, *then* $\mathbb{E}\big[M_i - M_{i-1} \mid E_2(\ell)\big] \geq 0$.

PROOF. In order to show that $\mathbb{E}[M_i - M_{i-1} \mid E_2(\ell)] \geq 0$, we will rewrite M_i and M_{i-1} as the sum of the workload processed before and after time $T_i + X$ in the event $E_2(\ell)$, and show that the conditional expectation of the difference in workloads is nonnegative both before and after time $T_i + X$.

On the one hand, the expected workload processed before time $T_i + X$ conditioned on the event $E_2(\ell)$ can be expressed as the sum of the expected elapsed service time of each replica at time $T_i + X$ conditioned on the event $E_2(\ell)$, that is

$$\mathbb{E}\left[\sum_{r=1}^{i-1} \min\big\{X(1+S_r), (T_i + X) - T_r\big\} \,\middle|\, E_2(\ell)\right]$$

when $i - 1$ replicas are created, and

$$\mathbb{E}\left[X + \sum_{r=1}^{i-1} \min\big\{X(1+S_r), (T_i + X) - T_r\big\} \,\middle|\, E_2(\ell)\right]$$

when i replicas are created. Consequently, the expected difference in workload processed before time $T_i + X$ is

$$\mathbb{E}[X \mid E_2(\ell)] \geq 0.$$

On the other hand, the expected workload processed after time $T_i + X$ conditioned on the event $E_2(\ell)$ can be expressed as the sum of the expected times that replicas spend between consecutive departures of replicas after

time $T_i + X$ conditioned on the event $E_2(\ell)$, multiplied by the number of replicas in service during that time, that is

$$\mathbb{E}\Bigg[(i-1-\ell)\left[D_{[\ell+1:i-1]} - (T_i + X)\right] + \sum_{r=\ell+1}^{k_n-1} (i-1-r)\left(D_{[r+1:i-1]} - D_{[r:i-1]}\right)\Bigg| E_2(\ell)\Bigg] \tag{B.4}$$

when $i-1$ replicas are created, and

$$\mathbb{E}\Bigg[(i-\ell)\left[D_{[\ell+1:i]} - (T_i + X)\right] + \sum_{r=\ell+1}^{k_n-1} (i-r)\left(D_{[r+1:i]} - D_{[r:i]}\right)\Bigg| E_2(\ell)\Bigg] \tag{B.5}$$

when i replicas are created. We will now argue that these two expressions are equal.

Consider the case where i replicas are made. Recall that the extended queue state process $\mathcal{Q}_n(\cdot)$, which keeps track of the size of the replicas and of their elapsed service times but not of their remaining service times, is Markov (Assumption 2.2). Moreover, recall that slowdowns are independent and exponentially distributed with mean $1/\mu$, and independent from the task size X (Assumption 3.1). Combining these two facts it follows that, conditioned on $E_2(\ell)$, the remaining service times of the $i-\ell$ replicas that are still in service at time $T_i + X$ are $X\tilde{S}_1, \ldots, X\tilde{S}_{i-\ell}$, where $\tilde{S}_1, \ldots, \tilde{S}_{i-\ell}$ are independent and exponentially distributed random variables with mean $1/\mu$, independent from X. Moreover, note that for $r = \ell+1, \ldots, k_n$, $D_{[r,i]} - (T_i + X)$ is the r-th order statistic of the remaining service times $X\tilde{S}_1, \ldots, X\tilde{S}_{i-\ell}$. It follows that, conditioned on $X = x$ and $E_2(\ell)$, the random variables $D_{[\ell+1:i]} - (T_i + X)$, and $D_{[r+1:i]} - D_{[r:i]}$ are exponential with rates $\mu(i-\ell)/x$

and $\mu(i-r)/x$, respectively. Combining this with Equation (B.5) we obtain

$$\begin{aligned}
&\mathbb{E}\left[(i-\ell)\left[D_{[\ell+1:i]}-(T_i+X)\right]+\sum_{r=\ell+1}^{k_n-1}(i-r)\left(D_{[r+1:i]}-D_{[r:i]}\right)\middle| E_2(\ell)\right] \\
&\quad=\mathbb{E}\left[\mathbb{E}\left[\left[(i-\ell)\left[D_{[\ell+1:i]}-(T_i+X)\right]\right.\right.\right. \\
&\qquad\qquad\left.\left.\left.+\sum_{r=\ell+1}^{k_n-1}(i-r)\left(D_{[r+1:i]}-D_{[r:i]}\right)\middle| X,\, E_2(\ell)\right]\middle| E_2(\ell)\right] \\
&\quad=\mathbb{E}\left[(i-\ell)\left[\frac{X}{\mu(i-\ell)}\right]+\sum_{r=\ell+1}^{k_n-1}(i-r)\left[\frac{X}{\mu(i-r)}\right]\middle| E_2(\ell)\right] \\
&\quad=\left(\frac{k_n-\ell}{\mu}\right)\mathbb{E}\big[X\,\big|\,E_2(\ell)\big].
\end{aligned}$$

Similarly, it can be checked that the expected workload processed after time T_i+X when only $i-1$ replicas are created (Equation (B.4)) is the same. □

Since the events E_1 and $E_2(0),\ldots,E_2(k_n-1)$ form a partition, lemmas B.1 and B.2 imply that

$$\begin{aligned}
\mathbb{E}[M_i-M_{i-1}] = \mathbb{E}[M_i-M_{i-1}\mid E_1]\mathbb{P}(E_1) \\
+\sum_{r=0}^{k_n-1}\mathbb{E}[M_i-M_{i-1}\mid E_2(r)]\mathbb{P}(E_2(r))
\end{aligned}$$

is nonnegative, for all i. Combining this with equations (B.2) and (B.3), the expected workload of a typical job is

$$\mathbb{E}[M_{n+k_n-1}]\geq k_n\left(1+\frac{1}{\mu}\right).$$

Finally, combining this with the stability condition in Equation (B.1)), we obtain the necessary condition for stability

$$\left(\frac{\lambda n}{k_n}\right)k_n\left(1+\frac{1}{\mu}\right)<n,$$

which is equivalent to

$$\lambda<\frac{1}{1+\frac{1}{\mu}}.$$

APPENDIX C: PROOF OF THEOREM 3.2

We consider a relaxation of the problem in which there are an infinite number of servers available, but there can only be up to n busy servers in expectation, in steady-state, and there can only be up to n replicas associated with the same job in service, at any point in time. Since this includes policies that have only up to n busy servers at all times, any lower bound for the delay in this relaxed setting is also a lower bound for the delay in the original setting. Thus, we shall prove a lower bound for this infinite-server relaxation. First, we will obtain a lower bound for the case where there are only finitely many task sizes, and then we generalize it to task sizes with general distributions via a comparison argument.

C.1. Finitely many task sizes. Consider the case where the task size X can only take values in a finite set $\mathcal{X}$, with $\mathbb{E}[X] = \eta$ (which is a further relaxation of the original assumption that $\mathbb{E}[X] = 1$). For each $x \in \mathcal{X}$, let $p(x) \triangleq \mathbb{P}(X = x)$ be the probability that a job has tasks of size x.

Let us fix a stable admissible policy, and consider the system in steady-state. For each $x \in \mathcal{X}$, let $\tilde{p}(x)$ be the expected number of servers (normalized by n) that are working on replicas of size x, in steady-state. In this setting, the proof is completed in three steps:

1. For each $x \in \mathcal{X}$, we obtain a lower bound for the expected time that a typical *replica* of size x spends in each phase, as a function of $\tilde{p}(x)$.
2. For each $x \in \mathcal{X}$, we show that the expected time that a typical *replica* of size x spends in each phase is smaller than or equal to expected time that a typical *job* with replicas of size x spends in the same phase. Combining this with the first step, and adding up the lower bounds for all phases and task sizes, we obtain a lower bound for the expected delay of a typical job as a function of $\{\tilde{p}(x) : x \in \mathcal{X}\}$.
3. We minimize the lower bound obtained in the previous step with respect to $\{\tilde{p}(x) : x \in \mathcal{X}\}$, which yields a lower bound on the expected delay of a typical job that only depends on the parameters of the system.

LEMMA C.1. *For every $x \in \mathcal{X}\backslash\{0\}$, and for $i = 1, \ldots, k_n$, we have*

$$\mathbb{E}\left[\overline{W}_n^{(i)} \middle| X = x\right] \geq \frac{x}{k_n\mu\left(\frac{\tilde{p}(x)}{\lambda p(x) x} - \frac{1}{\mu}\right) - \mu(i-1)},$$

where $\overline{W}_n^{(i)}$ is the time that a typical replica of size X spends in phase i.

PROOF. We first fix some $x \in \mathcal{X}\backslash\{0\}$, and introduce some notation. For $i = 0, \ldots, k_n$,

(i) let $N_n^{(i)}(x)$ be the steady-state number of replicas of size x, that are in phase i,
(ii) let $\lambda_n^{(i)}(x)$ be the arrival rate of replicas of size x, to phase i.

Recall that the extended queue state process $\mathcal{Q}_n(\cdot)$, which keeps track of the size of the replicas and of their elapsed service times but not of their remaining service times, is Markov (Assumption 2.2). Combining this with the fact that slowdowns are independent and exponential with mean $1/\mu$ (Assumption 3.1), and with the fact that a job either changes phases or leaves the system whenever a replica finishes its service, we see that the departure rate of jobs with tasks of size x from phase $i > 0$ is equal to

$$\frac{\mu}{x}\mathbb{E}\left[N_n^{(i)}(x)\right].$$

Furthermore, since all jobs go through all phases, then the arrival rate of jobs with tasks of size x to each phase is the same as the arrival rate of jobs with tasks of size x to the system, which is $\lambda n p(x)/k_n$. Since the system is in steady-state, the arrival and exit rates of jobs from each phase must be the same, and thus

$$\frac{\mu}{x}\mathbb{E}\left[N_n^{(i)}(x)\right] = \frac{\lambda p(x) n}{k_n},$$

for $i = 1, \ldots, k_n$. Equivalently, we have

$$\mathbb{E}\left[N_n^{(i)}(x)\right] = \frac{\lambda p(x) x n}{k_n \mu}, \tag{C.1}$$

for all $i = 1, \ldots, k_n$. In particular, this means that the expected number of replicas in phases 1 through k_n is independent from the policy and from the phase.

On the other hand, since $\tilde{p}(x)$ is the expected fraction of servers that are working on replicas of size x, we have

$$\sum_{i=0}^{k_n} \mathbb{E}\left[N_n^{(i)}(x)\right] \leq \tilde{p}(x) n.$$

The inequality above is an equality only if there are no wasteful replicas of size x, i.e., replicas that spend less than x units of time in service. Combining this with Equation (C.1), we obtain

$$\mathbb{E}\left[N_n^{(0)}(x)\right] \leq n\left(\tilde{p}(x) - \frac{\lambda p(x) x}{\mu}\right). \tag{C.2}$$

Moreover, recall that all useful replicas of size x spend exactly x units of time in phase 0, by definition. Thus,

$$\mathbb{E}\left[\overline{W}_n^{(0)} \middle| X = x\right] = x.$$

Combining this with Equation (C.2), and applying Little's law, we obtain

$$\lambda_n^{(0)}(x) = \frac{\mathbb{E}\left[N_n^{(0)}(x)\right]}{\mathbb{E}\left[\overline{W}_n^{(0)} \middle| X = x\right]} \leq n\left(\frac{\tilde{p}(x)}{x} - \frac{\lambda p(x)}{\mu}\right). \tag{C.3}$$

Recall that, for $i = 1, \ldots, k_n$, all the replicas that enter phase i had to enter phase 0 before. Furthermore, there are at least $i-1$ replicas associated with the same job that enter phase 0 but do not enter phase i (because they finished their service earlier). Combining these two facts, and using that the arrival rate of jobs with tasks of size x is $\lambda p(n)n/k_n$, we obtain

$$\lambda_n^{(i)}(x) \leq \lambda_n^{(0)}(x) - \frac{\lambda p(x)n}{k_n}(i-1), \tag{C.4}$$

for $i = 1, \ldots, k_n$. Applying Little's law once more, and using equations (C.3) and (C.4), we get

$$\begin{aligned}
\mathbb{E}\left[\overline{W}_n^{(i)} \middle| X = x\right] &= \frac{\mathbb{E}\left[N_n^{(i)}(x)\right]}{\lambda_n^{(i)}(x)} \\
&\geq \frac{\mathbb{E}\left[N_n^{(i)}(x)\right]}{\lambda_n^{(0)}(x) - \frac{\lambda p(x)n}{k_n}(i-1)} \\
&\geq \frac{x}{k_n\mu\left(\frac{\tilde{p}(x)}{\lambda p(x)x} - \frac{1}{\mu}\right) - \mu(i-1)}, \qquad \text{(C.5)}
\end{aligned}$$

for $i = 1, \ldots, k_n$. □

We have thus obtained a lower bound on the expected time that a typical replica of size x spends in phase i. However, we need to obtain a lower bound on the expected time that a typical *job* with tasks of size x spends in phase i. We prove that the former is smaller than or equal to the latter in the following claim.

LEMMA C.2. *For every $x \in \mathcal{X}\backslash\{0\}$, and for $i = 1, \ldots, k_n$, we have*

$$\mathbb{E}\left[W_n^{(i)} \middle| X = x\right] \geq \mathbb{E}\left[\overline{W}_n^{(i)} \middle| X = x\right],$$

where $W_n^{(i)}$ is the time that a typical job with tasks of size X spends in phase i.

PROOF. Recall that the time that a typical job spends in phase i is

$$W_n^{(i)} = \min\left\{T_1^{(i)} + XS_1^{(i)}, \ldots, T_n^{(i)} + XS_n^{(i)}\right\},$$

where $T_1^{(i)}, \ldots, T_n^{(i)}$ are the relative phase start times of replicas in the i-th phase, and $XS_1^{(i)}, \ldots, XS_n^{(i)}$ are the remaining service times of the replicas in the i-th phase. Moreover, recall that the extended queue state process $\mathcal{Q}_n(\cdot)$, which keeps track of the size of the replicas and of their elapsed service times but not of their remaining service times, is Markov (Assumption 2.2), and that slowdowns are independent and exponentially distributed with mean $1/\mu$, and independent from X (Assumption 3.1). Combining these two facts we get that $S_1^{(i)}, \ldots, S_n^{(i)}$ are independent and exponentially distributed with mean $1/\mu$, and independent from X. It follows that, conditioned on $X = x > 0$, the hazard rate of $W_n^{(i)}$ at t is equal to μ/x times the expected number of replicas in phase i associated with a typical job, t units of time after it has started its i-th phase, given that $W_n^{(i)} \geq t$. Namely, conditioned on $X = x > 0$, the hazard rate of $W_n^{(i)}$ at t is

$$h_{W_n^{(i)}}(t \mid x) = \frac{\mu}{x} \sum_{\ell=1}^{n} \mathbb{P}\left(T_\ell^{(i)} < t \,\middle|\, W_n^{(i)} > t,\, X = x\right).$$

Let $\overline{W}_n^{(i,r)}$ be the time that the r-th replica to enter phase i is in service, conditioned on having entered phase i at some point in time. In particular,

$$\mathbb{P}\left(\overline{W}_n^{(i,r)} > t \,\middle|\, X = x\right) = \mathbb{P}\Big(W_n^{(i)} - T_r^{(i)} > t \,\Big|\, W_n^{(i)} > T_r^{(i)},\, X = x\Big),$$

for all $t \geq 0$. Using the same argument used to obtain the hazard rate of $W_n^{(i)}$ we get that, conditioned on $X = x$, the hazard rate of $\overline{W}_n^{(i,r)}$ at t is

$$h_{\overline{W}_n^{(i,r)}}(t \mid x) = \frac{\mu}{x} \sum_{\ell=1}^{n} \mathbb{P}\left(T_\ell^{(i)} < t + T_r^{(i)} \,\middle|\, W_n^{(i)} > T_r^{(i)} + t,\, X = x\right),$$

It is easily checked that

$$h_{\overline{W}_n^{(i,r)}}(t \mid x) \geq h_{W_n^{(i)}}(t \mid x)$$

for all $t \geq 0$, and thus

$$\mathbb{E}\left[W_n^{(i)} \,\middle|\, X = x\right] \geq \mathbb{E}\left[\overline{W}_n^{(i,r)} \,\middle|\, X = x\right], \tag{C.6}$$

for all $r \geq 1$.

On the other hand, since the r-th replica of phase i starts its service only if $W_n^{(i)} > T_r^{(i)}$, it follows that the probability that a typical job with tasks of size x has at least r replicas associated with it in service during its i-th phase is

$$\mathbb{P}\left(W_n^{(i)} > T_r^{(i)} \middle| X = x\right).$$

Then, the expected number of replicas associated with a typical job with tasks of size x in phase i is

$$\sum_{k=1}^{n} \mathbb{P}\left(W_n^{(i)} > T_k^{(i)} \middle| X = x\right).$$

This means that the fraction of replicas of size x in phase i that are the r-th replica to start phase i in a typical job is

$$\frac{\mathbb{P}\left(W_n^{(i)} > T_r^{(i)} \middle| X = x\right)}{\sum_{k=1}^{n} \mathbb{P}\left(W_n^{(i)} > T_k^{(i)} \middle| X = x\right)}.$$

Therefore, the expected time that a typical replica in phase i is in service is equal to the weighted average

$$\mathbb{E}\left[\overline{W}_n^{(i)} \middle| X = x\right] = \sum_{r=1}^{n} \left[\frac{\mathbb{P}\left(W_n^{(i)} > T_r^{(i)} \middle| X = x\right)}{\sum_{k=1}^{n} \mathbb{P}\left(W_n^{(i)} > T_k^{(i)} \middle| X = x\right)}\right] \mathbb{E}\left[\overline{W}_n^{(i,r)} \middle| X = x\right].$$

Combining this with Equation (C.6), we obtain

$$\begin{aligned}\mathbb{E}\left[\overline{W}_n^{(i)} \middle| X = x\right] &\leq \sum_{r=1}^{n} \left[\frac{\mathbb{P}\left(W_n^{(i)} > T_r^{(i)} \middle| X = x\right)}{\sum_{k=1}^{n} \mathbb{P}\left(W_n^{(i)} > T_k^{(i)} \middle| X = x\right)}\right] \mathbb{E}\left[W_n^{(i)} \middle| X = x\right] \\ &= \mathbb{E}\left[W_n^{(i)} \middle| X = x\right],\end{aligned}$$

which concludes the proof of the lemma. □

Combining lemmas C.1 and C.2, we obtain

$$\begin{aligned}\mathbb{E}\big[W_n\big|X=x\big] &= \sum_{i=0}^{k_n}\mathbb{E}\left[W_n^{(i)}\middle|X=x\right]\\ &\geq x+\sum_{i=1}^{k_n}\mathbb{E}\left[\overline{W}_n^{(i)}\middle|X=x\right]\\ &\geq x\left(1+\sum_{i=1}^{k_n}\frac{1}{k_n\mu\left(\frac{\tilde{p}(x)}{\lambda p(x)x}-\frac{1}{\mu}\right)-\mu(i-1)}\right),\end{aligned}$$

for all $x\in\mathcal{X}\backslash\{0\}$. Moreover, combining this lower bound with the fact that the service time of any job with tasks of size 0 is 0, i.e. $\mathbb{E}[W_n(0)]=0$, and with the fact that the expected size of tasks is η, we obtain

$$\begin{aligned}\mathbb{E}[W_n] &= \sum_{x\in\mathcal{X}}p(x)\mathbb{E}\big[W_n\big|X=x\big]\\ &\geq \sum_{x\in\mathcal{X}\backslash\{0\}}xp(x)\left(1+\sum_{i=1}^{k_n}\frac{1}{k_n\mu\left(\frac{\tilde{p}(x)}{\lambda p(x)x}-\frac{1}{\mu}\right)-\mu(i-1)}\right)\\ &= \eta+\sum_{x\in\mathcal{X}\backslash\{0\}}\sum_{i=1}^{k_n}\frac{xp(x)}{k_n\mu\left(\frac{\tilde{p}(x)}{\lambda p(x)x}-\frac{1}{\mu}\right)-\mu(i-1)}. \qquad \text{(C.7)}\end{aligned}$$

At this point, we have a lower bound for the expected service time of a job, but it depends on the unknown quantities $\{\tilde{p}(x):x\in\mathcal{X}\backslash\{0\}\}$. In the following lemma, we obtain a lower bound that only depends on the system parameters λ, μ, η, and k_n, by minimizing the lower bound in Equation (C.7) with respect to the unknown quantities, over an appropriate domain.

LEMMA C.3. *We have*

$$\mathbb{E}[W_n]\geq\eta\left(1+\frac{1}{\mu}\sum_{i=1}^{k_n}\frac{1}{k_n\left(\frac{1}{\lambda\eta}-\frac{1}{\mu}\right)-i+1}\right).$$

PROOF. First, note that Equation (C.7) implies that

$$\mathbb{E}[W_n]\geq\min_{\tilde{p}}\left\{\eta+\sum_{x\in\mathcal{X}\backslash\{0\}}\sum_{i=1}^{k_n}\frac{xp(x)}{k_n\mu\left(\frac{\tilde{p}(x)}{\lambda p(x)x}-\frac{1}{\mu}\right)-\mu(i-1)}\right\},$$

where the minimization is over all "possible" quantities $\tilde{p} = \{\tilde{p}(x) : x \in \mathcal{X}\backslash\{0\}\}$. These largely depend on the dispatching/replication policy, but they are inherently constrained, as follows.

Recall that $\tilde{p}(x)$ is the expected number of servers (normalized by n) working on replicas of size x. Since there can only be up to n servers busy in expectation, we have

$$\sum_{x\in\mathcal{X}\backslash\{0\}} \tilde{p}(x) \leq 1. \tag{C.8}$$

Furthermore, since all jobs eventually leave the system, all jobs must have k_n associated replicas that actually finish their service. Thus, jobs with tasks of size x have at least k_n replicas associated with them that are in service at least x units of time. This implies that the arrival rate of replicas of size x that will be in service for at least x units of time has to be greater than or equal to k_n times the arrival rate of jobs with tasks of size x, i.e., we must have $\lambda_n^{(0)}(x) \geq \lambda p(x) n$. Combining this with Equation (C.3), we obtain

$$\tilde{p}(x) \geq \lambda p(x) x \left(1 + \frac{1}{\mu}\right), \tag{C.9}$$

for all $x \in \mathcal{X}\backslash\{0\}$. Using equations (C.8) and (C.9) to define the domain of our optimization problem in the variables $\{\tilde{p}(x) : x \in \mathcal{X}\backslash\{0\}\}$, we get that $\mathbb{E}[W_n]$ is lower bounded by

$$\begin{aligned}
\min_{\tilde{p}\in[0,1]^{|\mathcal{X}|-1}} \quad & \eta + \sum_{x\in\mathcal{X}\backslash\{0\}} \sum_{i=1}^{k_n} \frac{xp(x)}{k_n\mu\left(\frac{\tilde{p}(x)}{\lambda p(x)x} - \frac{1}{\mu}\right) - \mu(i-1)} \\
s.t. \quad & \tilde{p}(x) \geq \lambda p(x) x \left(1 + \frac{1}{\mu}\right), \qquad \forall x \in \mathcal{X}\backslash\{0\}, \\
& \sum_{x\in\mathcal{X}\backslash\{0\}} \tilde{p}(x) \leq 1.
\end{aligned}$$

Note that this is a finite-dimensional convex optimization problem. Taking the dual problem, and using the necessary condition of stability given in Theorem 3.1, it is easily checked that $\tilde{p}(x) = xp(x)/\eta$ minimizes the objective function. This results in the lower bound

$$\begin{aligned}
\mathbb{E}[W_n] &\geq \eta + \sum_{x\in\mathcal{X}} \sum_{i=1}^{k_n} \frac{xp(x)}{k_n\mu\left(\frac{1}{\lambda\eta} - \frac{1}{\mu}\right) - \mu(i-1)} \\
&= \eta \left(1 + \frac{1}{\mu} \sum_{i=1}^{k_n} \frac{1}{k_n\left(\frac{1}{\lambda\eta} - \frac{1}{\mu}\right) - i + 1}\right).
\end{aligned}$$

□

C.2. General task sizes. In this subsection we consider the case where the task size X has a general distribution with unit mean. Let us fix an admissible policy for the infinite-server relaxation. For each positive integer m, we consider a coupled system, as follows.

a) The arrival process is the same.
b) If a job with tasks of size X arrives to the original system, then a job with tasks of size
$$X^{(m)} \triangleq \min\left\{2^m,\ \frac{\lceil X2^m\rceil - 1}{2^m}\right\} \le X$$
arrives to the m-th coupled system.
c) For each job, replicas are always dispatched to idle servers, with the same relative phase start times as in the original system (unless the phase ends earlier due to the shrunk task sizes).
d) The slowdowns are the same as in the original system.

Since the replicas have the same relative phase start times (unless the phase ends earlier) and are subject to the same slowdowns, and since $X^{(m)} \le X$, we have
$$\min_{r=1,\dots,n}\left\{T_r^{(i)}(X) + X^{(m)}S_r^{(i)}\right\} \le \min_{r=1,\dots,n}\left\{T_r^{(i)}(X) + XS_r^{(i)}\right\},$$
for all $i = 1,\dots,k_n$, and for all m. In particular, this implies that the policy in the m-th coupled system has at most n servers busy in expectation, and that the expected delay of a job in the m-th coupled system is smaller than or equal to the one in the original system, i.e., that we have
$$\mathbb{E}\big[W_n\big] \ge \mathbb{E}\left[W_n^{(m)}\right], \tag{C.10}$$
for all m.

Finally, note that the replicas in the m-th coupled system can only take values in the finite set $\{l/2^m : 0 \le l \le 2^{2m}\}$. Then, we can use Lemma C.3 to obtain the lower bound
$$\mathbb{E}\left[W_n^{(m)}\right] \ge \mathbb{E}[X_m]\left(1 + \frac{1}{\mu}\sum_{i=1}^{k_n}\frac{1}{k_n\left(\frac{1}{\lambda\mathbb{E}[X_m]} - \frac{1}{\mu}\right) - i + 1}\right),$$
which holds for all $m \ge 1$. Combining this with Equation (C.10), we obtain
$$\mathbb{E}\big[W_n\big] \ge \sup_{m\ge 1}\left\{\mathbb{E}[X_m]\left(1 + \frac{1}{\mu}\sum_{i=1}^{k_n}\frac{1}{k_n\left(\frac{1}{\lambda\mathbb{E}[X_m]} - \frac{1}{\mu}\right) - i + 1}\right)\right\}. \tag{C.11}$$

We now proceed to compute this supremum. Note that

$$\mathbb{P}(X_m > x) = \mathbb{P}\left(X > \frac{\lfloor x2^m \rfloor + 1}{2^m}\right),$$

for all $x < 2^m$, and $\mathbb{P}(X_m > x) = 0$, for all $x \geq 2^m$. Using the fact that the sequence of events $\left\{X > \left(\lfloor x2^m \rfloor + 1\right)2^{-m}\right\}_{m\geq 1}$ is nondecreasing for any given $x \geq 0$, we have

$$\begin{aligned}\lim_{m\to\infty} \mathbb{P}(X_m \geq x) &= \lim_{m\to\infty} \mathbb{P}\left(X > \frac{\lfloor x2^m \rfloor + 1}{2^m}\right) \\ &= \mathbb{P}\left(\bigcup_{m=1}^{\infty} \left\{X > \frac{\lfloor x2^m \rfloor + 1}{2^m}\right\}\right) \\ &= \mathbb{P}\left(X > x\right),\end{aligned}$$

for all $x \geq 0$. Thus, the monotone convergence theorem implies

$$\lim_{m\to\infty} \mathbb{E}[X_m] = \lim_{m\to\infty} \int_0^{\infty} \mathbb{P}(X_m > x)dx = \int_0^{\infty} \mathbb{P}(X > x)dx = \mathbb{E}[X] = 1.$$

Combining this with Equation (C.11), we obtain

$$\mathbb{E}[W_n] \geq 1 + \frac{1}{\mu} \sum_{i=1}^{k_n} \frac{1}{k_n\left(\frac{1}{\lambda} - \frac{1}{\mu}\right) - i + 1},$$

which concludes the proof of Theorem 3.2.

APPENDIX D: PROOF OF THEOREM 3.4

We will be working under the assumption that $k_n = 1$. In this case, the FREC policy has the following convenient property.

LEMMA D.1. *Under the FREC policy for $k_n = 1$, all replicas associated with the same job start their service at the same time and leave the system at the same time.*

PROOF. Since $k_n = 1$, it is immediate that all replicas associated with the same job leave the system as the same time, that is, as soon as some replica completes service. Furthermore, it is also immediate that all replicas associated with jobs that find enough idle servers start their service at the same time.

We now focus on the replicas associated with jobs that are routed to the first subsystem, but do not find enough idle servers. Recall that when there are less than $\lceil r^* \rceil$ idle servers, replicas are created and dispatched to *all* servers in the subsystem, and they are not cancelled until $\lceil r^* \rceil$ replicas have started their service (or until one that did start its service, finishes). As a result, there can be at most one job with less than $\lceil r^* \rceil$ associated replicas in service. Combining this with the fact that $n^{(1)}$ is multiple of $\lceil r^* \rceil$, we conclude that all jobs have either $\lceil r^* \rceil$ or zero associated replicas in service, and that the number of busy servers is always a multiple of $\lceil r^* \rceil$. Thus, replicas that were dispatched to all servers start their service when the $\lceil r^* \rceil$ replicas associated with another job leave the system. Thus, all $\lceil r^* \rceil$ replicas associated with the same job start their service at the same time. The same argument applies for the second subsystem. □

Lemma D.1 implies that under the FREC policy for $k_n = 1$, we can think of each set of $\lceil r^* \rceil$ or $\lceil r^* \rceil - 1$ replicas associated with the same job as a single job being processed by a single server. Combined with the fact that queued replicas under the FREC policy wait for the first set of $\lceil r^* \rceil$ or $\lceil r^* \rceil - 1$ servers to become idle in a first-come first-serve fashion, it follows that the first and second subsystems behave as M/G/m_1 and M/G/m_2 queues, respectively, with

$$m_1 \triangleq \frac{n^{(1)}}{\lceil r^* \rceil} = \left\lfloor \lambda n \left(p_1 - 2n^{\alpha-1}\right)^+ \left(1 + \frac{1}{\mu \lceil r^* \rceil}\right) + \frac{\lambda n^\alpha}{\lceil r^* \rceil} \right\rfloor,$$

and

$$m_2 \triangleq \frac{n^{(2)}}{\lceil r^* \rceil - 1} = \left\lfloor \lambda n \left[1 - \left(p_1 - 2n^{\alpha-1}\right)^+\right] \left(1 + \frac{1}{\mu(\lceil r^* \rceil - 1)}\right) + \frac{\lambda n^\alpha}{\lceil r^* \rceil - 1} \right\rfloor.$$

In particular, since the arrival processes to both subsystems are obtained as independent thinnings of the original Poisson process of arrivals, they are Poisson processes as well, with rates

$$\lambda_n^{(1)} \triangleq \lambda n (p_1 - 2n^{\alpha-1})^+,$$

and

$$\lambda_n^{(2)} \triangleq \lambda n \left[1 - \left(p_1 - 2n^{\alpha-1}\right)^+\right].$$

Moreover, since all the replicas associated with the same job start and finish their service at the same time (cf. Lemma D.1), the service times of the jobs in the first and second subsystems are i.i.d. and distributed as $X(1 + \min\{S_1, \ldots, S_{\lceil r^* \rceil}\})$ and $X(1+\min\{S_1, \ldots, S_{\lceil r^* \rceil - 1}\})$, respectively. Thus, the

expected service times are

$$s^{(1)} \triangleq 1 + \frac{1}{\mu\lceil r^* \rceil}, \qquad \text{and} \qquad s^{(2)} \triangleq 1 + \frac{1}{\mu(\lceil r^* \rceil - 1)}.$$

Hence, the loads in the first and second subsystems are

$$\rho_n^{(1)} \triangleq \frac{\lambda_n^{(1)} s^{(1)}}{m_1} = \frac{\lambda n (p_1 - 2n^{\alpha-1})^+ \left(1 + \frac{1}{\mu\lceil r^* \rceil}\right)}{\left\lfloor \lambda n \left(p_1 - 2n^{\alpha-1}\right)^+ \left(1 + \frac{1}{\mu\lceil r^* \rceil}\right) + \frac{\lambda n^\alpha}{\lceil r^* \rceil} \right\rfloor} < 1,$$

and

$$\rho_n^{(2)} \triangleq \frac{\lambda_n^{(2)} s^{(2)}}{m_2} = \frac{\lambda n \left[1 - \left(p_1 - 2n^{\alpha-1}\right)^+\right] \left(1 + \frac{1}{\mu(\lceil r^* \rceil - 1)}\right)}{\left\lfloor \lambda n \left[1 - (p_1 - 2n^{\alpha-1})^+\right] \left(1 + \frac{1}{\mu(\lceil r^* \rceil - 1)}\right) + \frac{\lambda n^\alpha}{\lceil r^* \rceil - 1} \right\rfloor} < 1,$$

respectively. As a result, we have

$$1 - \rho_n^{(1)} \approx \frac{\frac{\lambda n^\alpha}{\lceil r^* \rceil}}{\left\lfloor \lambda n \left(p_1 - 2n^{\alpha-1}\right)^+ \left(1 + \frac{1}{\mu\lceil r^* \rceil}\right) + \frac{\lambda n^\alpha}{\lceil r^* \rceil} \right\rfloor} \in \Theta\left(n^{\alpha-1}\right),$$

and

$$1 - \rho_n^{(2)} \approx \frac{\frac{\lambda n^\alpha}{\lceil r^* \rceil - 1}}{\left\lfloor \lambda n \left[1 - (p_1 - 2n^{\alpha-1})^+\right] \left(1 + \frac{1}{\mu(\lceil r^* \rceil - 1)}\right) + \frac{\lambda n^\alpha}{\lceil r^* \rceil - 1} \right\rfloor} \in \Omega\left(n^{\alpha-1}\right),$$

Since the two subsystems behave exactly as M/G/m_1 and M/G/m_2 queues, their positive Harris recurrence is given by Theorem 2.2 and Corollary 2.8 in [4]. This guarantees the stability of the FREC policy.

For the service time, we have

$$\begin{aligned}\lim_{n\to\infty} \mathbb{E}[W_n^s] &= \lim_{n\to\infty} \left[1 + \frac{\left(p_1 - 2n^{\alpha-1}\right)^+}{\mu\lceil r^* \rceil} + \frac{1 - \left(p_1 - 2n^{\alpha-1}\right)^+}{\mu(\lceil r^* \rceil - 1)}\right] \\ &= 1 + \frac{p_1}{\mu\lceil r^* \rceil} + \frac{1 - p_1}{\mu(\lceil r^* \rceil - 1)}.\end{aligned}$$

It only remains to show that

$$\lim_{n\to\infty} \mathbb{E}[W_n^q] = 0.$$

Since both subsystems behave as M/G/m queues, and since there exists $\epsilon > 0$ such that $\mathbb{E}[X^{2+\epsilon}] < \infty$, Corollary 2 in [19] states that there exists a constant C_ϵ, independent from n, such that

$$\begin{aligned}\mathbb{E}[W_n^q] &\leq C_\epsilon \left[\frac{\left(p_1 - 2n^{\alpha-1}\right)^+}{\lambda_n^{(1)}\left(1-\rho_n^{(1)}\right)} + \frac{1-\left(p_1 - 2n^{\alpha-1}\right)^+}{\lambda_n^{(2)}\left(1-\rho_n^{(2)}\right)} \right] \\ &= C_\epsilon \left[\frac{1}{\lambda n\left(1-\rho_n^{(1)}\right)} + \frac{1}{\lambda n\left(1-\rho_n^{(2)}\right)} \right].\end{aligned}$$

Combining this with the fact that $1-\rho_n^{(1)} \in \Theta\left(n^{\alpha-1}\right)$ and $1-\rho_n^{(2)} \in \Omega\left(n^{\alpha-1}\right)$, we obtain

$$\lim_{n\to\infty} \mathbb{E}[W_n^q] \leq \lim_{n\to\infty} C_\epsilon \left[\frac{1}{\lambda n\left(1-\rho_n^{(1)}\right)} + \frac{1}{\lambda n\left(1-\rho_n^{(2)}\right)} \right] = 0,$$

which concludes the proof.

APPENDIX E: PROOF OF THEOREM 3.5

The proof is done in two steps. First, we show that the queueing delay converges to zero in probability (Subsection E.1), and then we show the convergence of the expected service time of a typical job to the desired constant (Subsection E.2).

E.1. Vanishing queueing delay. In this subsection we show that the large server pools defined by the DQ policy (i.e., the first $n^{(1)} - \lambda n^\alpha/2k_n$ and $n^{(2)} - \lambda n^\alpha/2k_n$ servers of the first and second subsystems, respectively) are stable, and that the queueing delay of a typical job converges to zero. We provide the complete proof of these facts for the first subsystem, with the second one being analogous.

For $i = 1, \ldots, k_n$, let $Q_i^{(1)}(\cdot)$ be the process describing the number of jobs in the large server pool of the first subsystem, for which less than i replicas associated with it have finished their service. Since all replicas associated with jobs sent to the large server pool start their service at the same time, $Q_i^{(1)}(\cdot)$ also describes the number of replicas in the large server pool of the first subsystem whose service time distributed as $X(1 + S_{[i:\lceil r^* k_n \rceil]})$, where $S_{[i:\lceil r^* k_n \rceil]}$ is the i-th order statistic of $\lceil r^* k_n \rceil$ slowdowns.

Recall that replicas arrive to the first subsystem in batches, as a Poisson process of rate

$$\lambda_n^{(1)} \triangleq \frac{\lambda n\left(p_{k_n} - 2n^{\alpha-1}\right)^+}{k_n} \in O\left(\frac{n}{k_n}\right),$$

and they are either dispatched to idle servers in the large server pool, or diverted to the small server pool (i.e., the last $\lambda n^\alpha/2k_n$ servers). As a result, for $i = 1, \ldots, k_n$, Proposition 2 in [28] implies that $Q_i^{(1)}(\cdot)$ is dominated by the queue length process, $\tilde{Q}_i^{(i)}(\cdot)$, of a queueing system where all the arrivals to the first subsystem are sent to idle servers. This corresponds to an M/G/∞ queue with Poison arrivals of rate $\lambda_n^{(1)}$, and i.i.d. jobs distributed as $X(1 + S_{[i:\lceil r^* k_n \rceil]})$, with expected service time

$$s_i^{(1)} \triangleq 1 + \mathbb{E}\left[S_{[i:\lceil r^* k_n \rceil]}\right] \in \Theta(1).$$

Moreover, for $i = 1, \ldots, k_n$, Proposition 2 in [28] implies that $\tilde{Q}_i^{(1)}(\cdot)$ is dominated by the queue length process, $\overline{Q}_i^{(1)}(\cdot)$, of an M/G/$n_i^{(1)}$ queue with

$$n_i^{(1)} \triangleq \left\lfloor \frac{\lambda n\left(p_{k_n} - 2n^{\alpha-1}\right)^+}{k_n}\left(1 + \mathbb{E}\left[S_{[i:\lceil r^* k_n \rceil]}\right]\right) + \frac{\lambda n^\alpha}{2k_n\lceil r^* k_n \rceil} - 1 \right\rfloor, \tag{E.1}$$

and the same arrivals and job sizes. It follows that, for $i = 1, \ldots, k_n$, we have

$$Q_i^{(1)}(t) \leq \overline{Q}_i^{(1)}(t), \tag{E.2}$$

for all $t \geq 0$. Since the load of the i-th M/G/$n_i^{(1)}$ queue is

$$\rho_{n,i}^{(1)} \triangleq \frac{\lambda_n^{(1)} s_i^{(1)}}{n_i^{(1)}} = \frac{\frac{\lambda n\left(p_{k_n} - 2n^{\alpha}\right)^+}{k_n}\left(1 + \mathbb{E}\left[S_{[i:\lceil r^* k_n \rceil]}\right]\right)}{\left\lfloor \frac{\lambda n\left(p_{k_n} - 2n^{\alpha}\right)^+}{k_n}\left(1 + \mathbb{E}\left[S_{[i:\lceil r^* k_n \rceil]}\right]\right) + \frac{\lambda n^\alpha}{2k_n\lceil r^* k_n \rceil} - 1 \right\rfloor} < 1, \tag{E.3}$$

then it is Harris recurrent (cf. [4]), for $i = 1, \ldots, k_n$. Combining this with Equation (E.2), we conclude that the original queues are also Harris recurrent. We denote the steady-state queue length processes of the original and of the larger queues by $Q_1^{(1)}, \ldots, Q_{k_n}^{(1)}$ and $\overline{Q}_1^{(1)}, \ldots, \overline{Q}_{k_n}^{(1)}$, respectively. Given the stochastic ordering of the original queue length processes, we have

$$Q_i^{(1)} \leq_{st} \overline{Q}_i^{(1)}, \tag{E.4}$$

for all $i = 1, \ldots, k_n$.

Let $W_n^{q,1}$ be the queueing delay of a typical job in the first subsystem. The vanishing of this delay is established in the following lemma.

LEMMA E.1. *We have*

$$\lim_{n\to\infty} \mathbb{P}\left(W_n^{q,1} > 0\right) = 0.$$

PROOF. Let $Q^{(1)}$ be the steady-state number of replicas present in the large server pool of the first subsystem. Since there is no queueing in the large server pool, $Q^{(1)}$ is also the steady-state number of busy servers in the large server pool of the first subsystem. Recall that under the DQ policy, a job can have positive queueing delay only if its diverted to the small server pool, which happens when there are less than $\lceil r^* k_n \rceil$ idle servers (or equivalently, more than $n^{(1)} - \lambda n^\alpha/(2k_n) - \lceil r^* k_n \rceil$ busy servers) in the large server pool at the time of the job's arrival. Since the arrivals are Poisson, the PASTA property implies that the steady-state probability of waiting is

$$\mathbb{P}\left(W_n^{q,1} > 0\right) \le \mathbb{P}\left(Q^{(1)} > n^{(1)} - \frac{\lambda n^\alpha}{2k_n} - \lceil r^* k_n \rceil\right). \tag{E.5}$$

Here there is an inequality because jobs sent to the small server pool might still experience zero queueing delay. Moreover, using the definition of the integers $n_1^{(1)}, \ldots, n_{k_k}^{(1)}$ given in Equation (E.1), and the fact that, for $i = 1, \ldots, k_n$, $Q_i^{(1)}$ is the steady-state number of jobs in the large server pool of the first subsystem, for which less than i replicas associated with it have finished their service, it can be checked that

$$n_1^{(1)} + \cdots + n_{k_n-1}^{(1)} + \Big(\lceil r^* k_n \rceil - k_n + 1\Big) n_{k_n}^{(1)} \le n^{(1)} - \frac{\lambda n^\alpha}{2k_n} - \lceil r^* k_n \rceil,$$

and that

$$Q^{(1)} = \left(\sum_{i=1}^{k_n-1} Q_i^{(1)}\right) + \Big(\lceil r^* k_n \rceil - k_n + 1\Big) Q_{k_n}^{(1)}.$$

Combining these two facts with Equation (E.4), it follows that

$$\begin{aligned}
&\mathbb{P}\left(Q^{(1)} > n^{(1)} - \frac{\lambda n^\alpha}{2k_n} - \lceil r^* k_n \rceil\right) \\
&\quad = \mathbb{P}\left(\left(\sum_{i=1}^{k_n-1} Q_i^{(1)}\right) + \big(\lceil r^* k_n \rceil - k_n + 1\big) Q_{k_n}^{(1)} > n^{(1)} - \frac{\lambda n^\alpha}{2k_n} - \lceil r^* k_n \rceil\right) \\
&\quad \leq \mathbb{P}\left(\left(\sum_{i=1}^{k_n-1} Q_i^{(1)}\right) + \big(\lceil r^* k_n \rceil - k_n + 1\big) Q_{k_n}^{(1)} > n_1^{(1)} + \cdots + n_{k_n-1}^{(1)} \right. \\
&\qquad\qquad \left. + \Big(\lceil r^* k_n \rceil - k_n + 1\Big) n_{k_n}^{(1)}\right) \\
&\quad \leq \mathbb{P}\left(\bigcup_{i=1}^{k_n} \left\{Q_i^{(1)} > n_i^{(1)}\right\}\right) \\
&\quad \leq \sum_{i=1}^{k_n} \mathbb{P}\left(Q_i^{(1)} > n_i^{(1)}\right) \\
&\quad \leq \sum_{i=1}^{k_n} \mathbb{P}\left(\overline{Q}_i^{(1)} > n_i^{(1)}\right). \qquad \text{(E.6)}
\end{aligned}$$

Recall that, for $i = 1, \ldots, k_n$, $\overline{Q}_i^{(1)}$ is the steady-state queue length of an M/G/$n_i^{(1)}$ queue with load $\rho_{n,i}^{(1)}$ (cf. Equation (E.3)), such that

$$1 - \rho_{n,i}^{(1)} \approx \frac{\frac{\lambda n^\alpha}{2k_n \lceil r^* k_n \rceil}}{\left\lfloor \frac{\lambda n \left(p_{k_n} - 2n^\alpha\right)^+}{k_n} \left(1 + \mathbb{E}\left[S_{[i:\lceil r^* k_n \rceil]}\right]\right) + \frac{\lambda n^\alpha}{2k_n \lceil r^* k_n \rceil} - 1 \right\rfloor} \in \Theta\left(\frac{n^{\alpha-1}}{k_n}\right).$$

Since there exists $\epsilon > 0$ such that $\mathbb{E}[X^{2+\epsilon}] < \infty$, Corollary 1 in [19] states that there exists a constant C_ϵ, independent from n and i, such that

$$\mathbb{P}\left(\overline{Q}_i^{(1)} > n_i^{(1)}\right) \leq \frac{C_\epsilon}{n_i^{(1)} \left(1 - \rho_{n,i}^{(1)}\right)^2}.$$

Combining this with the fact that $1 - \rho_{n,i}^{(1)} \in \Theta\left(n^{\alpha-1}/k_n\right)$, we obtain

$$\sum_{i=1}^{k_n} \mathbb{P}\left(\overline{Q}_i^{(1)} > n_i^{(1)}\right) \leq \sum_{i=1}^{k_n} \frac{C_\epsilon}{n_i^{(1)} \left(1 - \rho_{n,i}^{(1)}\right)^2} \in O\left(\frac{k_n^4}{n^{2\alpha-1}}\right).$$

Combining this with equations (E.5) and (E.6), we get that

$$\text{(E.7)} \qquad \mathbb{P}\left(W_n^{q,1} > 0\right) \leq \mathbb{P}\left(Q^{(1)} > n^{(1)} - \frac{\lambda n^\alpha}{2k_n} - \lceil r^* k_n \rceil\right) \in O\left(\frac{k_n^4}{n^{2\alpha-1}}\right).$$

Finally, since we assumed that $k_n \in O(n^\beta)$ and that $\alpha > (4\beta+2)/3$, then

$$\lim_{n\to\infty} \mathbb{P}\left(W_n^{q,1} > 0\right) = 0.$$

□

Similarly, if $W_n^{q,2}$ is the queueing delay of a typical job in the second subsystem, it can be shown that

$$\lim_{n\to\infty} \mathbb{P}\left(W_n^{q,2} > 0\right) = 0.$$

Finally, using the fact that a job is sent to the first subsystem with probability $\left(p_{k_n} - 2n^{\alpha-1}\right)^+$ and to the second subsystem with probability $1 - \left(p_{k_n} - 2n^{\alpha-1}\right)^+$, we get that the probability of a typical job having a positive queueing delay is

$$\begin{aligned}\lim_{n\to\infty} \mathbb{P}\left(W_n^q > 0\right) = \lim_{n\to\infty} &\left(p_{k_n} - 2n^{\alpha-1}\right)^+ \mathbb{P}\left(W_n^{q,1} > 0\right) \\ &+ \left[1 - \left(p_{k_n} - 2n^{\alpha-1}\right)^+\right] \mathbb{P}\left(W_n^{q,2} > 0\right) \\ = 0.&\end{aligned}$$

E.2. Convergence of the expected service time. In this subsection we show that the small server pools defined by the DQ policy are stable, and that the expected service time of a typical job converges to the desired constant. We provide the complete proof of these facts for the first subsystem, with the second one being analogous.

Recall that the arrival rate to the first subsystem is $\lambda n\left(p_{k_n} - 2n^{\alpha-1}\right)^+ / k_n$, and that jobs are only sent to the small server pool if there are less than $\lceil r^* k_n \rceil$ idle servers in the large server pool, which happens with probability

$$p_{small}^{(1)} \triangleq \mathbb{P}\left(Q^{(1)} > n^{(1)} - \frac{\lambda n^\alpha}{2k_n} - \lceil r^* k_n \rceil\right)$$

in steady-state. Then, the arrival rate of replicas to the small server pool is

$$\lambda_{small}^{(1)} \triangleq \frac{\lambda n\left(p_{k_n} - 2n^{\alpha-1}\right)^+ p_{small}^{(1)}}{k_n}.$$

Combining this with Equation (E.7), we have

$$\lambda_{small}^{(1)} \in O\left(\frac{k_n^3}{n^{2\alpha-2}}\right). \tag{E.8}$$

Since only k_n replicas are created for each job sent to the small server pool, no replicas are prematurely cancelled, and thus the expected service time of each replica is

$$s^{(1)} \triangleq 1 + \frac{1}{\mu}.$$

Combining this with Equation (E.8) and the fact that the small server pool has $\lambda n^\alpha / 2k_n$ servers, we get that the load of the small server pool of the first subsystem is

$$\rho_{small}^{(1)} \triangleq \frac{\lambda_{small}^{(1)} s^{(1)}}{\frac{\lambda n^\alpha}{2k_n}} \in O\left(\frac{k_n^4}{n^{3\alpha-2}}\right).$$

Since we assumed that $k_n \in O(n^\beta)$ and $\alpha > (4\beta+2)/3$, then the load of the small server pool converges to zero, and it is thus stable for all n large enough.

Let $\mathbb{E}\left[W_n^{s,1}\right]$ be the expected service time of a typical job in the first subsystem. We have the following convergence results.

LEMMA E.2. *If $k_n = k$ for all n, we have*

$$\lim_{n\to\infty} \mathbb{E}\left[W_n^{s,1}\right] = 1 + \frac{1}{\mu} \sum_{i=1}^{k} \frac{1}{\lceil r^* k \rceil - i + 1}.$$

If $k_n \to \infty$ as $n \to \infty$, we have

$$\lim_{n\to\infty} \mathbb{E}\left[W_n^{s,1}\right] = 1 + \frac{1}{\mu} \log\left(\frac{r^*}{r^* - 1}\right)$$

PROOF. Recall that under the DQ policy, all replicas associated with the same job sent to the large server pools start their service at the same time by construction. As a result, the service time of a job sent to the large server pool in first subsystem is distributed as the k_n-th order statistic of $X(1+S_1), \ldots, X(1+S_{\lceil r^* k_n \rceil})$. Thus, its expected service time is

$$1 + \frac{1}{\mu} \sum_{i=1}^{k_n} \frac{1}{\lceil r^* k_n \rceil - i + 1}.$$

On the other hand, note that the service time of any job is upper bounded by the sum of the service times of its replicas, by the definition of the service time of a job. Since jobs sent to the small server pool have only k_n replicas associated with them, then the expected service time of a job sent to the small server pool in the first subsystem, to be denoted by $s^{(1)}_{small}$, is upper bounded as follows:

$$s^{(1)}_{small} \leq k_n \left(1 + \frac{1}{\mu}\right). \tag{E.9}$$

Since a job is sent to the small server pool within the first subsystem with probability $p^{(1)}_{small}$, we see that the expected service time of a typical job in the first subsystem is

$$\mathbb{E}\left[W_n^{s,1}\right] \triangleq 1 + p^{(1)}_{small} s^{(1)}_{small} + \left(1 - p^{(1)}_{small}\right) \frac{1}{\mu} \sum_{i=1}^{k_n} \frac{1}{\lceil r^* k_n \rceil - i + 1}.$$

Combining this with equations (E.7) and (E.9), and the facts that $k_n \in O(n^\beta)$ and $\alpha > (5\beta + 1)/2$, we obtain

$$\lim_{n\to\infty} \mathbb{E}\left[W_n^{s,1}\right] = 1 + \frac{1}{\mu} \sum_{i=1}^{k} \frac{1}{\lceil r^* k \rceil - i + 1}$$

for the case where $k_n = k$ for all n, and

$$\lim_{n\to\infty} \mathbb{E}\left[W_n^{s,1}\right] = 1 + \frac{1}{\mu} \log\left(\frac{r^*}{r^* - 1}\right)$$

for the case where $k_n \to \infty$ as $n \to \infty$. □

Similarly, if $\mathbb{E}\left[W_n^{s,2}\right]$ is the expected service time of a typical job in the second subsystem, it can be shown that

$$\lim_{n\to\infty} \mathbb{E}\left[W_n^{s,2}\right] = 1 + \frac{1}{\mu} \sum_{i=1}^{k} \frac{1}{\lceil r^* k \rceil - i}$$

for the case where $k_n = k$ for all n, and

$$\lim_{n\to\infty} \mathbb{E}\left[W_n^{s.,2}\right] = 1 + \frac{1}{\mu} \log\left(\frac{r^*}{r^* - 1}\right)$$

for the case where $k_n \to \infty$ as $n \to \infty$.

Finally, using the fact that a job is sent to the first subsystem with probability $\left(p_{k_n} - 2n^{\alpha-1}\right)^+$ and to the second subsystem with probability $1 - \left(p_{k_n} - 2n^{\alpha-1}\right)^+$, we get that the expected service time of a typical job in the system is

$$\mathbb{E}\left[W_n^s\right] = \left(p_{k_n} - 2n^{\alpha-1}\right)^+ \mathbb{E}\left[W_n^{s,1}\right] + \left[1 - \left(p_{k_n} - 2n^{\alpha-1}\right)^+\right] \mathbb{E}\left[W_n^{s,2}\right],$$

and thus

$$\lim_{n\to\infty} \mathbb{E}\left[W_n^s\right] = 1 + \frac{1}{\mu} \sum_{i=1}^{k} \left(\frac{p_k}{\lceil r^* k \rceil - i + 1} + \frac{1 - p_k}{\lceil r^* k \rceil - i} \right)$$

for the case where $k_n = k$ for all n, and

$$\lim_{n\to\infty} \mathbb{E}\left[W_n^s\right] = 1 + \frac{1}{\mu} \log\left(\frac{r^*}{r^* - 1} \right)$$

for the case where $k_n \to \infty$ as $n \to \infty$.

APPENDIX F: PROOF OF LEMMA 4.1

For Block policies, all replicas associated with the same job start their service at the same time. As a result, the expected service time of job with tasks of size x for which exactly r replicas start their service is

$$1 + \mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big].$$

Moreover, the expected total server time that each job requires in this case is

$$(r - k_n)x\Big(1 + \mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big]\Big) + \sum_{i=1}^{k_n} x\Big(1 + \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big]\Big).$$

Indeed, the second term is the sum of the smallest k_n service times, which correspond to the service times of replicas that do finish their service. The first term is the total server time of the $(r - k_n)$ replicas that do not finish their service, but are nevertheless in service for $x\big(1 + \mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big]\big)$ units of time.

Averaging these expressions over the number of replicas created (according to the distribution $p(x)$) and over the possible task sizes (according to the task size distribution $\mathbb{P}_X$), the expected service time of a typical job is

$$\int_0^\infty x \sum_{r=k_n}^{\infty} p_r(x)\Big(1 + \mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big]\Big) d\mathbb{P}_X(x),$$

and the expected server time that a typical job requires is

$$\int_0^\infty x \sum_{r=k_n}^{\infty} p_r(x) \Bigg[r + (r-k_n)\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] + \sum_{i=1}^{k_n} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] \Bigg] d\mathbb{P}_X(x). \tag{F.1}$$

On the other hand, by Little's law, the expected number of busy servers in steady-state is equal to the arrival rate of jobs ($\lambda n/k_n$) times the expected server time (in Equation (F.1)). Since this must be less than or equal to the total number of servers (n), we have

$$\frac{\lambda}{k_n} \int_0^\infty x \sum_{r=k_n}^{\infty} p_r(x) \Bigg[r + (r-k_n)\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] + \sum_{i=1}^{k_n} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] \Bigg] d\mathbb{P}_X(x) \leq 1.$$

APPENDIX G: PROOF OF LEMMA 4.2

Note that the condition

$$\lambda \leq \frac{1}{1 + \frac{1}{\mu}},$$

implies that the problem is feasible ($p_{k_n}(x) = 1$ for all x is a feasible solution). Moreover, the objective function is lower bounded by 1. These two facts imply that the infimum is finite.

Let us now define the function $I : \mathbb{R} \to \mathbb{R}$ as the optimal objective value of the perturbed problem:

$$\begin{aligned} I(u) \triangleq \inf_{p \in \mathcal{P}_{k_n}} \quad & \int_0^\infty x \sum_{r=k_n}^{\infty} p_r(x) \Big(1 + \mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big]\Big) d\mathbb{P}_X(x) \\ s.t. \quad & \frac{\lambda}{k_n} \int_0^\infty x \sum_{r=k_n}^{\infty} p_r(x) \Big[r + (r-k_n)\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] \\ & \qquad + \sum_{i=1}^{k_n} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] \Big] d\mathbb{P}_X(x) \leq 1 + u. \end{aligned}$$

Moreover, consider the function $g : \mathcal{P}_{k_n} \to \mathbb{R}$, defined by

$$g(p) \triangleq \frac{\lambda}{k_n} \int_0^\infty x \sum_{r=k_n}^\infty p_r(x) \Bigg[r + (r-k_n)\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] + \sum_{i=1}^{k_n} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] \Bigg] d\mathbb{P}_X(x) - 1.$$

It is easy to check that

$$0 \in \operatorname{core}\big[g(\mathcal{P}_{k_n}) + \mathbb{R}_+\big],$$

which implies that $I(\cdot)$ is relatively continuous at 0, and thus its subdifferential at zero is non-empty. Combining this with the fact that the infimum is finite, we use Theorem 4 in [22] to conclude that the infimum is attained.

APPENDIX H: PROOF OF THEOREM 4.3

We start with a technical result about the optimization problem.

LEMMA H.1. *Consider the function $I : \mathbb{R} \to \mathbb{R}$ defined as the optimal objective value of the perturbed problem:*

$$\begin{aligned} I(u) \triangleq \inf_{p \in \mathcal{P}_{k_n}} \quad & \int_0^\infty x \sum_{r=k_n}^\infty p_r(x) \Big(1 + \mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big]\Big) d\mathbb{P}_X(x) \\ s.t. \quad & \frac{\lambda}{k_n} \int_0^\infty x \sum_{r=k_n}^\infty p_r(x) \Big[r + (r-k_n)\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] \\ & \qquad + \sum_{i=1}^{k_n} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] \Bigg] d\mathbb{P}_X(x) \le 1 + u. \end{aligned}$$

Then we have that $I(\cdot)$ is relative continuous at 0, and we have the strong duality

$$\begin{aligned} I(0) = \sup_{y \ge 0} \inf_{p \in \mathcal{P}_{k_n}} \quad & \int_0^\infty x \sum_{r=1}^\infty p_r(x) \Bigg[1 - y + \frac{y\lambda r}{k_n} + \mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] \\ & \Big(1 + \frac{y\lambda}{k_n}(r-k_n)\Big) + \frac{y\lambda}{k_n} \sum_{i=1}^{k_n} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big] \Bigg] d\mathbb{P}_X(x). \end{aligned} \tag{H.1}$$

Moreover, $I(0)$ is attained by the dual problem at points (p^*, y^*) with $y^* > 0$.

PROOF. The relative continuity of $I(\cdot)$, the strong duality, and the attainability of the optimal value by the dual are obtained using the same arguments as in the proof of Lemma 4.2, given in Appendix G.

Moreover, since the inequality constraint in the problem is clearly active, i.e., since we have $0 \notin \partial I(0)$, then $y^* > 0$ for every optimal dual solution (p^*, y^*). □

Since we have strong duality (Lemma H.1), we focus on the solution of the dual problem:

$$\sup_{y\geq 0}\inf_{p\in\mathcal{P}_{k_n}} \int_0^\infty x \sum_{r=1}^\infty p_r(x)\left[1-y+\frac{y\lambda r}{k_n}+\mathbb{E}\big[S_{[k_n:r]}(x)\big]\left(1+\frac{y\lambda}{k_n}(r-k_n)\right)\right.$$
$$\left.+\frac{y\lambda}{k_n}\sum_{i=1}^{k_n}\mathbb{E}\big[S_{[i:r]}(x)\big]\right] d\mathbb{P}_X(x).$$

Furthermore, since Lemma H.1 states that the supremum with respect to y is attained at points with $y > 0$, we can take the supremum over all $y > 0$ and obtain the same solutions. Moreover, since there are no constraints involving multiple values of x at the same time, we can take the infimum for each x separately, inside the integral. Using these two facts, we obtain the equivalent problem:

$$\sup_{y>0} \int_0^\infty x \inf_{p(x)\in P_{k_n}} \left\{\sum_{r=1}^\infty p_r(x)\left[1-y+\frac{y\lambda r}{k_n}+\mathbb{E}\big[S_{[k_n:r]}\,\big|\,X=x\big]\right.\right.$$
$$\left.\left.\left(1+\frac{y\lambda}{k_n}(r-k_n)\right)+\frac{y\lambda}{k_n}\sum_{i=1}^{k_n}\mathbb{E}\big[S_{[i:r]}\,\big|\,X=x\big]\right]\right\} d\mathbb{P}_X(x).$$

We now explore the properties of the solutions. Fix $x \geq 0$ and $y > 0$, and consider the problem

$$\inf_{p(x)\in P_{k_n}}\left\{\sum_{r=1}^\infty p_r(x)\left[1-y+\frac{y\lambda r}{k_n}+\mathbb{E}\big[S_{[k_n:r]}\,\big|\,X=x\big]\left(1+\frac{y\lambda}{k_n}(r-k_n)\right)\right.\right.$$
$$\left.\left.+\frac{y\lambda}{k_n}\sum_{i=1}^{k_n}\mathbb{E}\big[S_{[i:r]}\,\big|\,X=x\big]\right]\right\}.$$

Since this minimization is a linear program over an infinite-dimensional simplex, any optimal solution, if it exists, will be concentrated on the set of

indices

$$\arg\min_{r\geq k_n}\left\{\frac{y\lambda r}{k_n}+\mathbb{E}\big[S_{[k_n:r]}\,\big|\,X=x\big]\left(1+\frac{y\lambda}{k_n}(r-k_n)\right)+\frac{y\lambda}{k_n}\sum_{i=1}^{k_n}\mathbb{E}\big[S_{[i:r]}\,\big|\,X=x\big]\right\}.$$

We now proceed to explore how these indices depend on x and k_n. Let us define the function $f_{x,y}:\mathbb{N}\to\mathbb{R}$ such that

$$f_{x,y}(r)\triangleq\frac{y\lambda r}{k_n}+\mathbb{E}\big[S_{[k_n:r]}\,\big|\,X=x\big]\left(1+\frac{y\lambda}{k_n}(r-k_n)\right)+\frac{y\lambda}{k_n}\sum_{i=1}^{k_n}\mathbb{E}\big[S_{[i:r]}\,\big|\,X=x\big].$$

For $r\geq 2$, we have

$$\begin{aligned}
&f_{x,y}(r)-f_{x,y}(r-1)\\
&\quad=\frac{y\lambda}{k_n}\left(1+\sum_{i=1}^{k_n}\mathbb{E}\big[S_{[i:r]}\,\big|\,X=x\big]-\mathbb{E}\big[S_{[i:r-1]}\,\big|\,X=x\big]\right)\\
&\qquad+\Big(\mathbb{E}\big[S_{[k_n:r]}\,\big|\,X=x\big]-\mathbb{E}\big[S_{[k_n:r-1]}\,\big|\,X=x\big]\Big)\left[1+\frac{y\lambda}{k_n}(r-k_n)\right]\\
&\qquad\qquad+\frac{y\lambda}{k_n}\mathbb{E}\big[S_{[k_n:r-1]}\,\big|\,X=x\big].
\end{aligned}$$

Since

$$\lim_{x\to\infty}\Big(\mathbb{E}\big[S_{[i:r]}\,\big|\,X=x\big]-\mathbb{E}\big[S_{[i:r-1]}\,\big|\,X=x\big]\Big)=0$$

and

$$\lim_{x\to\infty}\mathbb{E}\big[S_{[k_n:r-1]}\,\big|\,X=x\big]=\frac{1}{\mu},$$

then $f_{x,y}(r)-f_{x,y}(r-1)>0$, for all x large enough. Thus, $r^*(x,y)=k_n$ for all x large enough. This proves part (i) of the theorem.

On the other hand, we have

$$\begin{aligned}
f_{x,y}(r)&\leq\frac{y\lambda r}{k_n}+\mathbb{E}\big[S_{[k_n:r]}\,\big|\,X=x\big]\left(1+\frac{y\lambda}{k_n}(r-k_n)\right)\\
&\qquad\qquad+\frac{y\lambda}{k_n}\sum_{i=1}^{k_n}\mathbb{E}\big[S_{[k_n:r]}\,\big|\,X=x\big]\\
&=\frac{y\lambda r}{k_n}+\mathbb{E}\big[S_{[k_n:r]}\,\big|\,X=x\big]\left(1+\frac{y\lambda r}{k_n}\right).
\end{aligned}$$

Since $\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] - 1/\mu$ is either increasing or decreasing in x, we have that either

$$\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] \leq \mathbb{E}\big[S_{[k_n:r]} \mid X = 0\big],$$

or

$$\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] \leq \lim_{x\to\infty} \mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] = \frac{1}{\mu}.$$

Either way, we have

$$\mathbb{E}\big[S_{[k_n:r]} \,\big|\, X = x\big] \leq \max\left\{\mathbb{E}\big[S_{[k_n:r]} \mid X = 0\big], \frac{1}{\mu}\right\},$$

for all r. Moreover, since $S_{[k_n:r]}$ is the k_n-th order statistic of r i.i.d. random variables with mean $1/\mu$, there exists $\underline{r} \in O(k_n)$ such that $\mathbb{E}\big[S_{[k_n:r]} \mid X = 0\big] \leq 1/\mu$ for all $r \geq \underline{r}$. Thus,

$$f_{x,y}(r) \leq \frac{y\lambda r}{k_n} + \frac{1}{\mu}\left(1 + \frac{y\lambda r}{k_n}\right) \triangleq \overline{f_{x,y}}(r),$$

for all $r \geq \underline{r}$. Combining this with the fact that

$$f_{x,y}(r) \geq \frac{y\lambda r}{k_n} \triangleq \underline{f_{x,y}}(r),$$

we see that $f_{x,y}$ is sandwiched between the two affine functions $\underline{f_{x,y}}$ and $\overline{f_{x,y}}$, for all $r \geq \underline{r}$. It follows that the minimum of $f_{x,y}$ is achieved at indices upper bounded by the largest positive integer $\overline{r^*_{k_n}}$ such that

$$\overline{f_{x,y}}(\underline{r}) \geq \underline{f_{x,y}}(\overline{r^*_{k_n}}).$$

Equivalently, for all $y > 0$, we have

$$\overline{r^*_{k_n}} \leq \underline{r} + \frac{1}{\mu}\left(\frac{k_n}{y\lambda} + \underline{r}\right) \in O(k_n).$$

Finally, since this holds for all $y > 0$, in particular it holds for the optimal $y^* > 0$, and part (ii) of the theorem is proved.

APPENDIX I: PROOF OF THEOREM 4.5

We first present a simple result on the expectation of the minimum of i.i.d. random variables.

LEMMA I.1. *Let $S_1, S_2, \ldots$ be a sequence of nonnegative and non constant i.i.d. random variables. Then,*

$$g(r) \triangleq \mathbb{E}\big[\min\{S_1, \ldots, S_r\}\big]$$

is a strictly convex function, i.e., we have

$$g(r) < \frac{g(r-1) + g(r+1)}{2},$$

for all $r \geq 2$.

PROOF. We have

$$\begin{aligned}
&\mathbb{E}[\min\{S_1, \ldots, S_{r+1}\}] \\
&\quad = \mathbb{E}\big[\min\{S_1, \ldots, S_r\}\big] - \mathbb{E}\Big[\big(\min\{S_1, \ldots, S_r\} - S_{r+1}\big)\mathbb{1}_{\{\min\{S_1,\ldots,S_r\} > S_{r+1}\}}\Big].
\end{aligned}$$

This means that

$$\begin{aligned}
g(r+1) - g(r) &= -\mathbb{E}\Big[\big(\min\{S_1, \ldots, S_r\} - S_{r+1}\big)\mathbb{1}_{\{\min\{S_1,\ldots,S_r\} > S_{r+1}\}}\Big] \\
&= -\mathbb{E}\Big[\mathbb{E}\Big[\big(\min\{S_1, \ldots, S_r\} - S_{r+1}\big)\mathbb{1}_{\{\min\{S_1,\ldots,S_r\} > S_{r+1}\}} \mid S_{r+1}\Big]\Big].
\end{aligned}$$

Since $S_1, \ldots, S_{r+1}$ are independent, we have

$$\begin{aligned}
&\mathbb{E}\Big[\mathbb{E}\Big[\big(\min\{S_1, \ldots, S_r\} - S_{r+1}\big)\mathbb{1}_{\{\min\{S_1,\ldots,S_r\} > S_{r+1}\}} \mid S_{r+1}\Big]\Big] \\
&\qquad = \int_0^\infty \mathbb{E}\Big[\big(\min\{S_1, \ldots, S_r\} - s\big)\mathbb{1}_{\{\min\{S_1,\ldots,S_r\} > s\}}\Big] dF_{S_{r+1}}(s).
\end{aligned}$$

Note that, for all $s \geq 0$, the integrand is a nonincreasing function of r. Moreover, since S_{r+1} is not constant, the integrand is a decreasing function of r for a set of values of s with positive probability (with respect to $F_{S_{r+1}}$). It follows that $g(r+1) - g(r)$ is a decreasing function of r. As a result,

$$[g(r+1) - g(r)] - [g(r) - g(r-1)] > 0,$$

and thus

$$g(r) < \frac{g(r-1) + g(r+1)}{2}.$$

□

For the case $k_n = 1$, the optimization problem defined by equations (4.1) and (4.2) simplifies to the following.

$$\begin{aligned} \inf_{p\in\mathcal{P}_1} \quad & \int_0^\infty x \sum_{r=1}^\infty p_r(x)\Big(1+\mathbb{E}\big[S_{[1:r]} \,\big|\, X=x\big]\Big) d\mathbb{P}_X(x) \\ s.t. \quad & \lambda \int_0^\infty x \sum_{r=1}^\infty r p_r(x)\Big(1+\mathbb{E}\big[S_{[1:r]} \,\big|\, X=x\big]\Big) d\mathbb{P}_X(x) \le 1. \end{aligned}$$

Since we have strong duality (Lemma H.1), we focus on the solution of the dual problem:

$$\sup_{y\ge 0} \inf_{p\in\mathcal{P}_1} \quad \int_0^\infty x \sum_{r=1}^\infty p_r(x)\Big[\Big(1+\mathbb{E}\big[S_{[1:r]} \,\big|\, X=x\big]\Big)(1+y\lambda r) - y\Big] d\mathbb{P}_X(x)$$

Furthermore, since Lemma H.1 states that the supremum with respect to y is attained at points with $y > 0$, we can take the supremum over all $y > 0$ and obtain the same solutions. Moreover, since there are no constraints involving multiple values of x at the same time, we can take the infimum for each x separately, inside the integral. Using these two facts, we obtain the equivalent problem:

$$\sup_{y>0} \quad \int_0^\infty x \inf_{p(x)\in P_1} \left\{ \sum_{r=1}^\infty p_r(x)\Big[\Big(1+\mathbb{E}\big[S_{[1:r]} \,\big|\, X=x\big]\Big)(1+y\lambda r) - y\Big] \right\} d\mathbb{P}_X(x).$$

We now explore the properties of the solutions. Fix $x \ge 0$ and $y > 0$, and consider the problem

$$\inf_{p(x)\in P_1} \quad \sum_{r=1}^\infty p_r(x)\Big[\Big(1+\mathbb{E}\big[S_{[1:r]} \,\big|\, X=x\big]\Big)(1+y\lambda r) - y\Big].$$

Since this is a linear program over a simplex, the solutions are concentrated on the set of indices

$$\arg\min_{r\ge 1}\Big[\Big(1+\mathbb{E}\big[S_{[1:r]} \,\big|\, X=x\big]\Big)(1+y\lambda r) - y\Big].$$

We now proceed to explore how these indices depend on x. Let us define the function $f_{x,y} : \mathbb{N} \to \mathbb{R}$ by

$$f_{x,y}(r) \triangleq \Big(1+\mathbb{E}\big[S_{[1:r]} \,\big|\, X=x\big]\Big)(1+y\lambda r) - y.$$

First, note that Lemma I.1 states that $\mathbb{E}\big[S_{[1:r]} \,\big|\, X = x\big]$ is a strictly convex function in r. Furthermore, Assumption 4.3 states that $r\mathbb{E}\big[S_{[1:r]} \,\big|\, X = x\big]$ is also a convex function in r. As a result, $f_{x,y}(r)$ is a strictly convex function in r. Moreover, since $y > 0$, we have

$$\lim_{r\to\infty} f_{x,y}(r) = \infty.$$

Combining this with the fact that $f_{x,y}$ is strictly convex, we can conclude that the minimum of $f_{x,y}$ is achieved in either one point, or in two consecutive points. This proves part (i) of the theorem.

On the other hand, for $r \geq 2$, consider

$$f_{x,y}(r) - f_{x,y}(r-1) = (1 + y\lambda r)\Big(\mathbb{E}\big[S_{[1:r]} \,\big|\, X = x\big] - \mathbb{E}\big[S_{[1:r-1]} \,\big|\, X = x\big]\Big) + y\lambda\Big(1 + \mathbb{E}\big[S_{[1:r-1]} \,\big|\, X = x\big]\Big).$$

By our assumptions on the slowdowns, $\mathbb{E}\big[S_{[1:r-1]} \,\big|\, X = x\big]$ and $\mathbb{E}\big[S_{[1:r]} \,\big|\, X = x\big] - \mathbb{E}\big[S_{[1:r-1]} \,\big|\, X = x\big]$ are increasing in x, for all $r \geq 2$. As a result, $f_{x,y}(r) - f_{x,y}(r-1)$ is increasing in x, for all $r \geq 2$. Since $f_{x,y}$ is convex, it follows that the minimum of $f_{x,y}$ is achieved at integers $r^*(x,y)$, which are nonincreasing with x. This proves part (iii) of the theorem.

Finally, note that the minimum of $f_{x,y}(\cdot)$ is achieved at two consecutive integers $r^*_{x,y}$ and $r^*_{x,y} + 1$ only if $f_{x,y}(r^*_{x,y} + 1) - f_{x,y}(r^*_{x,y}) = 0$. Since this difference is strictly increasing in x, the minimum can only be achieved at a certain pair of consecutive integers for only one value of x, call it $\hat{x}$. Thus, if $\mathbb{P}(X = \hat{x}) = 0$, optimal solutions are equal almost everywhere to a solution that is concentrated only on $r^*_{x,y}$. This proves part (ii) of the theorem.

APPENDIX J: PROOF OF THEOREMS 4.6 AND 4.7

Note that the SB-FREC and SB-DQ policies are almost the same as the FREC and DQ policies introduced in Subsection 3.2. The main difference is that there are more than two subsystems (but still finitely many of them), and that the routing of incoming jobs to the subsystems depends on the task sizes of the incoming jobs. Thus, while it is clear that the SB-DQ policy is a Block policy by construction, the fact that the SB-FREC policy is a Block policy is established using the same argument that was used to establish that the FREC policy is a Block policy (Lemma D.1).

Moreover, the convergence of the queueing delays to zero under the SB-FREC and SB-DQ policies follow the same arguments given in the proofs

of theorems 3.4 and 3.5 in sections D and E, respectively. It only remains to prove that the expected service times converge to the stated limits.

Consider the function $I : \mathbb{R} \to \mathbb{R}$ given by

$$\begin{aligned} I(u) = \inf_{p \in \mathcal{P}_k} \quad & \int_0^\infty x \sum_{r=k}^\infty p_r(x) \Big(1 + \mathbb{E}\big[S_{[k:r]} \,\big|\, X = x\big]\Big) d\mathbb{P}_X(x) \\ s.t. \quad & \frac{\lambda}{k} \int_0^\infty x \sum_{r=k}^\infty p_r(x) \Big[r + (r-k)\mathbb{E}\big[S_{[k:r]} \,\big|\, X = x\big] \\ & \qquad + \sum_{i=1}^{k_n} \mathbb{E}\big[S_{[i:r]} \,\big|\, X = x\big]\Big] d\mathbb{P}_X(x) \leq 1 + u. \end{aligned}$$

By construction, the expected service time of a typical job under the SB-FREC and SB-DQ policies is

$$\mathbb{E}[W_n^s] = \int_0^\infty x \sum_{r=k}^\infty p_r^{(n)}(x) \Big(1 + \mathbb{E}\big[S_{[k:r]} \,\big|\, X = x\big]\Big) d\mathbb{P}_X(x),$$

Moreover, the definition of $p^{(n)}$ implies that

$$\mathbb{E}[W_n^s] = I\big(-n^{\alpha-1}\big). \tag{J.1}$$

On the other hand, the definition of p^* implies that

$$\int_0^\infty x \sum_{r=k}^\infty p_r^*(x) \Big(1 + \mathbb{E}\big[S_{[k:r]} \,\big|\, X = x\big]\Big) d\mathbb{P}_X(x) = I(0). \tag{J.2}$$

Finally, combining equations (J.1) and (J.2) with the fact that $I(\cdot)$ is continuous around 0 (Lemma H.1), we obtain

$$\begin{aligned} \lim_{n\to\infty} \mathbb{E}[W_n^s] &= \lim_{n\to\infty} I\big(-n^{\alpha-1}\big) \\ &= I(0) \\ &= \int_0^\infty x \sum_{r=k}^\infty p_r^*(x) \Big(1 + \mathbb{E}\big[S_{[k:r]} \,\big|\, X = x\big]\Big) d\mathbb{P}_X(x), \end{aligned}$$

which proves the convergence of the expected service times.

REFERENCES

[1] Ananthanarayanan, G., Ghodsi, A., Shenker, S. and Stoica, I. (2012). Why let resources idle? Aggressive cloning of jobs with Dolly. In *Proceedings of HotCloud*.

[2] Ananthanarayanan, G., Ghodsi, A., Shenker, S. and Stoica, I. (2013). Effective Straggler Mitigation: Attack of the Clones. In *Proceedings of NSDI*.

[3] Anton, E., Ayesta, U., Jonckheere, M. and Verloop, I. M. (2019). On the stability of redundancy models. arXiv:1903.04414.

[4] Asmussen, S. (2003). *Applied Probability and Queues*. Springer.

[5] Berg, B., Dorsman, J.-P. and Harchol-Balter, M. (2017). Towards Optimality in Parallel Job Scheduling. *Proceedings of the ACM on Measurement and Analysis of Computing Systems - SIGMETRICS*.

[6] Borst, S., Boxma, O., Groote, J. F. and Mauw, S. (2003). Task allocation in a multi-server system. *Journal of Scheduling* **6** 423–436.

[7] Chen, S., Sun, Y., Kozat, U. C. and Huang, L. (2014). When queueing meets coding: Optimal-latency data retrieving scheme in storage clouds. In *Proceedings of INFOCOM*.

[8] Chu, C.-T., Kim, S. K., Lin, Y.-A., Yu, Y., Bradski, G., Ng, A. Y. and Olukotun, K. (2007). Map-Reduce for Machine Learning on Multicore. In *Proceedings of NIPS*.

[9] Duffy, K. R. and Shneer, S. (2019). MDS coding is better than replication for job completion times. arXiv:1907.11052.

[10] Gardner, K., Harchol-Balter, M., Hyytia, E. and Righter, R. (2017). Scheduling for Efficiency and Fairness in Systems with Redundancy. *Performance Evaluation* **116** 1–25.

[11] Gardner, K., Harchol-Balter, M. and Scheller-Wolf, A. (2016). A better model for job redundancy: Decoupling server slowdown and job size. In *Proceedings of MASCOTS*.

[12] Gardner, K., Harchol-Balter, M., Scheller-Wolf, A., Velednitsky, M. and Zbarsky, S. (2017). Redundancy-d: The Power of d Choices for Redundancy. *Operations Research* **65**.

[13] Gardner, K., Hyytia, E. and Righter, R. (2019). A little redundancy goes a long way: Convexity in redundancy systems. *Performance Evaluation* **131** 22–42.

[14] Joshi, G., Liu, Y. and Soljanin, E. (2012). Coding for Fast Content Download. In *Proceedings of Allerton*.

[15] Joshi, G., Soljanin, E. and Wornell, G. (2017). Efficient Redundancy Techniques for Latency Reduction in Cloud Systems. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)* **2**.

[16] Koole, G. and Righter, R. (2008). Resource Allocation in Grid Computing. *Journal of Scheduling* **11** 163–173.

[17] Kumar, A. and Shorey, R. (1993). Performance analysis and scheduling of stochastic fork-join jobs in a multicomputer system. *IEEE Transactions of Parallel Distributed Systems* **4** 1147–1164.

[18] Lee, K., Shah, N. B., Huang, L. and Ramchandran, K. (2017). The MDS Queue: Analysing the Latency Performance of Erasure Codes. *IEEE Transactions on Information Theory* **63** 2822–2842.

[19] Li, Y. and Goldberg, D. (2017). Simple and explicit bounds for multi-server queues with universal $\frac{1}{1-\rho}$ (and better) scaling. arXiv:1706.04628.

[20] Liang, G. and Kozat, U. C. (2014). TOFEC: Achieving optimal throughput-delay trade-off of cloud storage using erasure codes. In *Proceedings of INFOCOM*.

[21] Lu, Y., Xie, Q., Kliot, G., Geller, A., Larus, J. R. and Greenberg, A. (2011). Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services. *Performance Evaluation* **68** 1056-1071.

[22] Mitter, S. (2008). Convex Optimization in Infinite Dimensional Spaces. In *Blondel V.D., Boyd S.P., Kimura H. (eds) Recent Advances in Learning and Control. Lecture Notes in Control and Information Sciences, vol 371.* Springer, London.

[23] Nelson, R., Towsley, D. and Tantawi, A. N. (1988). Performance Analysis of Parallel Processing Systems. *IEEE Transactions of Software Engineering* **14** 532–540.

[24] Poloczek, F. and Ciucu, F. (2016). Contrasting Effects of Replication in Parallel Systems: From Overload to Underload and Back. In *Proceedings of SIGMETRICS*.

[25] Rizk, A., Poloczek, F. and Ciucu, F. (2016). Stochastic bounds in Fork-Join queueing systems under full and partial mapping. *Queueing Systems* **83** 261–297.

[26] Shah, N. B., Lee, K. and Ramchandran, K. (2016). When Do Redundant Requests Reduce Latency? *IEEE Transactions on Communications* **64** 715-722.

[27] Shah, V., Bouillard, A. and Baccelli, F. (2017). Delay comparison of delivery and coding policies in data clusters. In *Proceedings of Allerton*.

[28] Shanthikumar, J. G. and Yao, D. D. (1989). Stochastic Monotonicity in General Queueing Networks. *Journal of Applied Probability* **26** 413–417.

[29] Sun, Y., Koksal, C. E. and Shroff, N. B. (2017). On Delay-Optimal Scheduling in Queueing Systems with Replications. arXiv:1603:07322.

[30] Thomasian, A. (2014). Analysis of Fork/Join and Related Queueing Systems. *ACM Computing Surveys* **47** 1–71.

[31] Vulimiri, A., Godfrey, P. B., Mittal, R., Sherry, J., Ratnasamy, S. and Shenker, S. (2013). Low latency via redundancy. In *Proceedings of CoNEXT*.

[32] Wang, D., Joshi, G. and Wornell, G. (2014). Efficient task replication for fast response times in parallel computation. In *Proceedingss of SIGMETRICS*.

[33] Wang, D., Joshi, G. and Wornell, G. (2015). Using Straggler Replication to Reduce Latency in Large-scale Parallel Computing. In *Proceedingss of SIGMETRICS*.

[34] Wang, W., Harchol-Balter, M., Jiang, H., Scheller-Wolf, A. and Srikant, R. (2018). Delay Asymptotics and Bounds for Multi-Taks Parallel Jobs. *ACM SIGMETRICS Performance Evaluation Review* **46** 2–7.

Martin Zubeldia
Department of Mathematics & Computer Science
Eindhoven University of Technology
5600 MB Eindhoven, Netherlands.
E-mail: m.zubeldia.suarez@tue.nl