Load Balancing with Job-Size Testing: Performance Improvement or Degradation?

JONATHA ANSELMI, Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, France
JOSU DONCEL, University of the Basque Country, UPV/EHU, Spain

In the context of decision making under explorable uncertainty, scheduling with testing is a powerful technique used in the management of computer systems to improve performance via better job-dispatching decisions. Upon job arrival, a scheduler may run some *testing algorithm* against the job to extract some information about its structure, e.g., its size, and properly classify it. The acquisition of such knowledge comes with a cost because the testing algorithm delays the dispatching decisions, though this is under control. In this paper, we analyze the impact of such extra cost in a load balancing setting by investigating the following questions: does it really pay off to test jobs? If so, under which conditions? Under mild assumptions connecting the information extracted by the testing algorithm in relationship with its running time, we show that whether scheduling with testing brings a performance degradation or improvement strongly depends on the traffic conditions, system size and the coefficient of variation of job sizes. Thus, the general answer to the above questions is non-trivial and some care should be considered when deploying a testing policy. Our results are achieved by proposing a load balancing model for scheduling with testing that we analyze in two limiting regimes. When the number of servers grows to infinity in proportion to the network demand, we show that job-size testing actually degrades performance unless short jobs can be predicted reliably almost instantaneously *and* the network load is sufficiently high. When the coefficient of variation of job sizes grows to infinity, we construct testing policies inducing an arbitrarily large performance gain with respect to running jobs untested.

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

Additional Key Words and Phrases: Load balancing, dispatching policies, size-based routing, scheduling with testing, explorable uncertainty

ACM Reference Format:
Jonatha Anselmi and Josu Doncel. 2023. Load Balancing with Job-Size Testing: Performance Improvement or Degradation?. In . ACM, New York, NY, USA, 27 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The efficient management of multi-server distributed systems strongly depends on the ability to properly classify incoming jobs, as this information can lead to better scheduling decisions. In the typical scenario, the exact structure of the incoming jobs is not known in advance but some information about their size or running time may be extracted by running an auxiliary *testing algorithm*. This algorithm takes as input the description of a job and retrieves information by performing operations of various levels of complexity; Finally, such information is exploited to make a job-size prediction and to dispatch the given job to the right resources for processing with the objective of minimizing some performance criteria such as the mean delay.

In principle, the computational resources that can be dedicated to the testing algorithm are unbounded because it is well known in computer science that the "halting problem", i.e., the problem of determining whether a computer program will finish running or continue to run forever from a description of an arbitrary computer program and an input, is undecidable. On the other hand, while the execution of the testing algorithm comes with a price (because the job dispatching decisions are delayed), it may provide valuable classification information that eventually pays off. The investigation of such exploration-exploitation trade-off under explorable uncertainty, referred in the literature with the term "scheduling with testing", has recently gained traction in scheduling theory [2, 19, 28] and is the focus of this paper.

1.1 Motivation

Scheduling with testing relies on the idea of possibly running a *test* against each submitted job. Tests aim at extracting information about the structure of jobs such as their sizes or running times *before* their actual processing. This technique finds applications in a wide range of domains that we describe in the following, though our main motivation is the application to High Performance Computing (HPC) distributed systems [30, 33].

In HPC systems, users submit jobs to the platform over time possibly with an estimation of their running times, or equivalently job sizes, and a central scheduler relies on this information to make a prediction about their sizes and finally dispatch them to the right computational resources [33, 35]. It is well known that user-provided running time estimations are not accurate [10, 21, 30, 33], hence the idea of improving the estimation by running tests. In practice, typical job running times are "highly variable" in the sense that they can be of the order of minutes, hours or days, and these are not known to the scheduler in advance; see, e.g., [23, Section 3] for neuroscience applications.

The main objective consists in developing job dispatching strategies that minimize a performance measure of interest such as the the mean delay or *makespan*, and this strongly relies on the ability to reduce the interference between long and short jobs. Towards this purpose, scheduling with testing may be a viable option that can be performed at several levels depending on the available computational resources [35]. Here, the output of a test is a guess (subject to error) of the exact size of the given job. With an increasing level of complexity, a test may consist in guessing a job size by *i)* just extracting the user-provided estimation of the job execution time (though this is inaccurate as remarked in [33, 35]), *ii)* analyzing the job input parameters [23], *iii)* analyzing the job source code (e.g., looking at the Kolmogorov complexity) [31], *iv)* running a code optimizer [12, 18], or *v)* running a prediction toolbox constructed from previous data via machine learning mechanisms [21, 30, 33]. For linear algebra operations, just looking at input parameters such as the dimension of the input matrix may be enough to extract useful information [23]. In fact, if the matrix dimension is small, then the job is most likely short. On the other hand, the contrapositive may not hold: if the job is large, then it is not necessarily long because the matrix may be sparse, and in this case one may consider a more sophisticated testing strategy digging in the structure of the matrix more deeply and therefore with an increased testing cost. Other examples include machine learning or image processing jobs where the input parameters may be the number of iterations to train the underlying machine learning model or the number of pixels of the given image, respectively. In this context, it is not well understood whether the extra cost induced by scheduling with testing brings a performance benefit or degradation (see Section 1.2 for further details), and this open question motivates our work.

Besides HPC systems, scheduling with testing is also employed in several other settings. In communication networks for instance, a testing algorithm may be a compression algorithm aimed at reducing the size, and thus the transmission time, of the given file to transmit, and here a compression cost must clearly be paid [34]; note that an already compressed file is uncompressible but this knowledge is not available beforehand. In wireless networks, a testing algorithm may be an algorithm that selects the channel with the optimal signal-to-noise ratio before the actual transmission takes place. In maintenance or medical environments, a diagnosis or a test can be carried out to determine the exact processing time of a job, and this information is used in turn to prioritize and efficiently allocate resources [3, 28, 29]. We refer the reader to [1, 18, 19] for more detailed discussions about applications of scheduling with testing.

1.2 Related Work

Scheduling with testing falls in the broad field of *optimization under explorable uncertainty*. More specifically, it describes a setting where jobs have unknown processing times that can be explored by running a test for some time with a scheduler dispatching them tested or untested on a number of computational resources (servers) to minimize some performance measure of interest. From a theoretical point of view, this problem has been primarily investigated on a *single* server [1, 17–19, 28] with the objective of minimizing the sum of completion times. A step further has been recently taken in [2], where the authors consider the natural generalization of the problem to *multiple* identical parallel servers. In these works, the authors provide algorithms that decide the amount of testing to be performed and demonstrate their effectiveness by means of competitive-ratio analyses. These are meant to compare the value of an algorithm with respect to an optimal off-line solution. In the case of multiple servers, the SBS algorithm developed in [2] is 3-competitive with respect to the makespan, i.e., the maximum load on any server, provided that testing times are uniform.

An important assumption underlying the references above is that they focus on static settings with a finite number of jobs. In this paper however, we focus on multiple identical parallel servers as in [2] but consider a stochastic and dynamic setting where an *infinite* number of jobs joins the system over time following an exogenous stochastic process. This allows us to develop testing strategies that can guarantee stability of the scheduler in the sense of positive Harris recurrence of the underlying Markov process. Moreover, existing works assume that the exact job sizes are revealed after testing [1, 2, 17, 19], while we will consider in Section 2.2 a more general stochastic model connecting the testing times and the amount of revealed information.

To the best of our knowledge, our work is the first to investigate scheduling with testing in a queueing-theoretic load-balancing setting. Nonetheless, similar approaches have been considered in the queueing literature. Most of the existing works about load balancing (or equivalently job dispatching among a set of parallel queues) assume that jobs are dispatched to servers instantly upon their arrival with no cost, i.e., disregarding the impact of testing; see [15] for a recent survey. Size-based routing (or load balancing) and the Task Assignment Guessing Size (TAGS) algorithm [7, 8, 24] are two job-dispatching strategies that can be interpreted as particular instances of scheduling with testing in a queueing-theoretic setting. However, these rely on assumptions that make them structurally different with respect to our approach. Let us briefly discuss these two approaches. Size-based routing operates under the assumption that the scheduler knows the exact size of each job upon its arrival *without* testing; see, e.g., [5, 9, 25, 36]. The knowledge of exact job sizes allows one to develop dispatching strategies preventing long jobs from blocking the execution of several short jobs behind, especially in first-in first-out systems. It is well known that this approach reduces the mean delay induced by classical load balancing algorithms such as join-the-shortest-queue provided that job sizes are variable enough [4, 25, 27]. Size-based routing is a degenerate case of scheduling with testing where the cost of testing is zero. The TAGS strategy, proposed in [24], operates as follows; see also [8]. Upon arrival, a job is dispatched to server one. If its execution takes less than t_1 time units, it leaves the system, otherwise, it is killed after t_1 time units and re-executed from scratch at server two. Here, if its execution takes less than $t_2 > t_1$ time units, it leaves the system, otherwise, it is killed after t_2 time units and re-executed from scratch at server three, and the process repeats until the last server is found, where the job is forced to complete. TAGS implements a form of job-size testing strategy because at the n-th stage, the hosting server knows that its executing time is at least t_{n-1}. Specifically, the test coincides with the execution of the job itself for a limited amount of time. The drawback of TAGS is its reduced stability region (due to the fact that jobs need to be re-executed from scratch upon re-routing) and the fact that job migration across servers often implies a prohibitively expensive communication overhead. In fact, this holds true in HPC systems as typical jobs involve a large amount of data.

Finally, Markov decision processes may be used to identify optimal testing strategies. However, this approach is prohibitively expensive from a computational point of view due to the curse of dimensionality, as the size of the underlying state space is exponential in the number of servers.

1.3 Contribution

We propose a Markovian framework for scheduling with testing in a load balancing setting. Specifically, an infinite sequence of jobs joins the system over time through a central scheduler (or dispatcher), which is in charge of routing them to one out of N parallel servers. Each server has its own queue, processes jobs at speed one and operates under the first-come first-served scheduling discipline. The scheduler is aware of the probability distribution of job sizes but it can also rely on a testing algorithm to extract additional information about the exact size of each job, and this is exploited to route the job to some server with the objective of minimizing the mean delay. Therefore, two mechanisms dictate the dynamics of the central scheduler:

i) a *testing* policy, which defines the amount of time allocated to the execution of the testing algorithm,
ii) a *dispatching* (or load balancing) policy, which defines how jobs are dispatched to the N servers.

To take into account the extra delay induced by the testing policy, we model the scheduler as an M/M/1 queue where the service times coincide with the testing times, and we will focus on testing policies that guarantee stability. Though the existing literature on load balancing is vast, our framework is the first to consider a load balancing system with testing policies in a stochastic and dynamic setting.

Under a general assumption connecting the testing algorithm running time and the amount of revealed information (see Assumption 1), we analyze the resulting mean waiting time of jobs in two orthogonal limiting regimes: i) a limiting regime where the system size N grows to infinity in proportion to the traffic demand while keeping the network load constant, and ii) a limiting regime where the coefficient of variation of job sizes grow to infinity. Importantly, the amount of resources at the central scheduler does not scale in either case. These regimes, widely considered in queueing theory to approximate dynamics in the pre-limit, are justified because real HPC systems are large and because common empirical studies of computer systems show that realistic probability distributions of job sizes are heavy tailed; e.g., [22, Chapter 9] and [14, 26].

Our main results identify the conditions under which job-size testing is effective (or not) to improve the performance of a system that runs jobs untested. From a practical standpoint, here the idea is that a monitor runs in the background and makes decisions about testing times depending on the detected condition. Our findings show that these conditions strongly depend on the traffic conditions, system size and the coefficient of variation of job sizes, which implies that some attention should be paid when deploying a testing policy. To quickly summarize our results:

- In the large system limit ($N \to \infty$), we show that job-size testing is *not* worth unless each job brings enough information that can be retrieved instantly *and* the network load is sufficiently high (Theorems 1 and 2).

- For any system size N, it is possible to design testing policies such that their performance benefit is arbitrarily large if and only if the job sizes are variable enough (Section 4.2).

Let us elaborate on these insights.

When $N \to \infty$, the mean testing time necessarily converges to zero as otherwise the stability condition at the scheduler would not be satisfied. This leads us to investigate dynamics on a right-neighborhood of zero. Here, we show in Theorem 1 that the derivative of the mean waiting time with respect to the mean testing time is strictly increasing if the amount of information that the testing algorithm can retrieve instantly upon job arrival is null. The latter condition appears natural and can be rephrased more formally as "the random variable modelling *the prediction* of the job size obtained with a zero testing time is independent of the random variable modeling the job size". In some applications however, predictions about *short* jobs are reliable and obtained almost instantaneously by just looking at input parameters, while predictions about long jobs are subject to errors; as commented above, for most of matrix operations a job can be regarded as short if the size of the given matrix is small while the contrapositive does not necessarily hold true. This fact leads us to consider the "No False Small" scenario; see Section 3.3 for a precise definition. Within this scenario and on a right-neighborhood of zero, we show in Theorem 2 that the derivative of the mean waiting time with respect to the mean testing time is strictly negative if the traffic demand is sufficiently large or if the probability of having short jobs and exact predictions increases sufficiently fast with the mean testing time.

When the job sizes follow a heavy tailed distribution where the coefficient of variation grows to infinity, which is the case of practical interest, the mean testing time can be designed in a way to control the mean testing cost. Within this regime and within the testing time design choice given in (17), we provide in Theorem 3 conditions ensuring that the efficiency ratio, i.e., the ratio between the mean waiting time obtained with and without testing, converges to zero. These conditions include the "Independent Predictions for Zero Testing Time" and "No False Small" scenarios discussed above, and identify the settings where job-size testing, to our opinion, should be applied. In contrast, if job sizes are not variable enough, the "universal" lower bound in Corollary 1 immediately implies that job-size testing is inefficient.

Finally, our conclusions are supported by numerical results obtained within job size distributions drawn from a neuroscience application [23].

1.4 Organization

The rest of this work is organized as follows. In Section 2, we propose a new framework for job-size testing and dispatching. This is then analyzed in the limit where the system size grows to infinity in Section 3 and in the limit where the degree of variability of job sizes grows to infinity in Section 4. Then, Section 5 presents empirical results, and Section 6 draws the conclusions of our work.

2 LOAD BALANCING WITH JOB-SIZE TESTING

Summarized in Figure 1, we describe a framework for load balancing with job-size testing and define the performance measures of interest.

2.1 Architecture

We consider a service system composed of N servers, a sequence of *jobs* and a *scheduler* (or dispatcher). Each of the N servers has its own infinite-room queue and processes jobs at unit rate following the first-come first-served scheduling discipline. In the following, the terms "server" and "queue" will be used interchangeably.

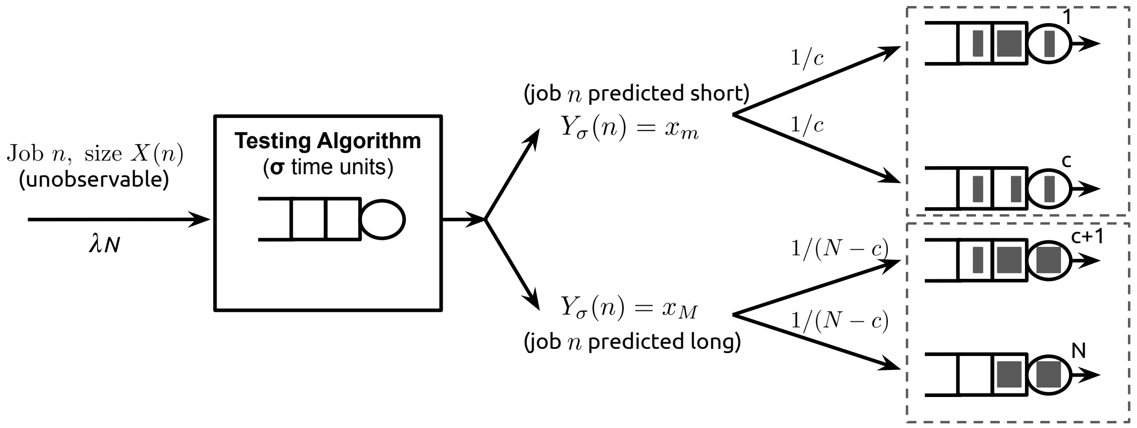

Fig. 1. Architecture of the proposed model for load balancing with job-size testing. It is assumed that $c \in \mathbb{N}$.

2.1.1 Jobs. Jobs join the system over time following a Poisson process with rate $\Lambda := \lambda N$ and each of them leaves the system after service completion at its designated server.

The n-th arriving job has a size given by random variable $X(n)$ and this is the amount of work required by the n-th job on one server, or equivalently its service time at any server since servers operate at unit rate. We assume that the sequence $(X(n))_n$ is independent, identically distributed and also independent of the arrival process. For stability, i.e., positive Harris recurrence of the underlying Markov process, we assume that $\rho := \lambda \mathbb{E}[X(n)] < 1$.

We also assume that jobs are classified as either *long*, i.e., of size x_M, or *short*, i.e., of size x_m, with $0 < x_m < x_M < \infty$. Thus, $\mathbb{P}(X(n) = x_m) + \mathbb{P}(X(n) = x_M) = 1$ for all n. We will be interested in applying our results to the case where job sizes have a "heavy-tailed" distribution in the sense that

$$\mathbb{P}(X = x_M) = \alpha\, x_M^{-\beta}, \quad \alpha > 0,\ \beta > 0, \tag{1}$$

with x_M "large". Here, the tail of X decays polynomially and we will be particularly interested in the case where $\beta < 1$ as this implies high variability when x_M grows.

Let us justify the two-point job size distribution assumption:

- In the context of HPC systems, common workloads are composed of jobs whose size can be clustered in two groups, "long" and "short", which yields bimodal distributions [13, 23, 37]; see also [11, 16, 27]. Several reasons induce such dichotomy and these may depend on the application at hand. For instance, in neuroscience applications, users submit jobs together with images that may contain either little or a lot of information, with a gap of several orders of magnitude [23]. Here, our two-point distribution assumption is meant to simplify and approximate such bimodal distribution.
- From a theoretical point of view, a two-point distribution will be enough both to exhibit the issues and to capture the key properties underlying job-size testing for load balancing. In the context of scheduling with testing, other works rely on this assumption to develop *worst-case* analyses [17]. In fact, we remark that the distribution that maximizes the variance among the distributions with bounded support concentrates on the two extreme points of the support. Thus, in this sense it yields the worst-case scenario as the mean delay (defined in (4)) strongly depends on the variance of job sizes.

A generalization of our theoretical results to general job-size probability distribution is out of the scope of this paper. However, in Section 5.2, we numerically show that our framework is robust to perturbations of two-point job-size distributions.

2.1.2 Scheduler. When jobs arrive, they immediately join a scheduler, which is in charge of dispatching each job to exactly one queue. In order to make such decisions, it is allowed to execute a *testing algorithm* against each job and using its own resources. This algorithm extracts some information that is exploited to make a prediction about its actual size. The *testing time*, i.e. the amount of time dedicated to the execution of the testing algorithm, is deterministic and under the control of the system manager. We assume that the same testing time is applied to all jobs, say $\sigma \geq 0$.

We model the scheduler as an $M/M/1$ queue with arrival rate Λ and mean service time σ. This is meant to model congestion and to capture the impact of the runtime phenomena happening during the execution of the testing algorithm, which stochastically perturb the service times,

For stability, we choose σ such that $\Lambda\sigma < 1$. Using the Pollaczek–Khinchine formula [6], the mean delay (sojourn time) at the scheduler is

$$\frac{\sigma}{1 - \Lambda\sigma}, \tag{2}$$

which can also be interpreted as the mean delay of an $M/G/1$ processor-sharing queue.

We let $Y_\sigma(n)$ denote the $\{x_m, x_M\}$-valued random variable associated to the prediction of the true (unknown) size, $X(n)$, of the n-th arriving job when the testing algorithm is executed for σ time units. We assume that the prediction of the n-th job $Y_\sigma(n)$ can only depend on $X(n)$ and σ. With a slight abuse of notation, (X, Y_σ) denotes an auxiliary random vector having the same distribution of the job size and prediction pair $(X(n), Y_\sigma(n))$.

The following remark summarizes the information available at the scheduler.

Remark 1. *The scheduler knows λ, N and the distribution of the job size X. For all job n, it does not know the exact job size $X(n)$ but can execute a testing algorithm for σ time units to obtain a prediction, $Y_\sigma(n)$, of the size of job n. Finally, the scheduler knows the joint probability mass function of (X, Y_σ) for all $\sigma \geq 0$, which in practice can be learned from the data.*

Thus, the joint distribution of job sizes and predictions is assumed to be obtained *in advance* by means of a parameter estimation phase. This is common in queueing theory.

2.1.3 Dispatching Policy. For job dispatching, we assume that there exists a real number $c \in (0, N)$ such that all jobs that are *predicted* as short ($Y_\sigma(n) = x_m$) resp. long ($Y_\sigma(n) = x_M$) are sent to servers $1, \ldots, \lfloor c \rfloor + 1$ resp. $\lfloor c \rfloor + 1, \ldots, N$. We will refer to c as *cutoff server*. Conditioned on $Y_\sigma = x_m$ resp. $Y_\sigma = x_M$, a job is sent to server $i = 1, \ldots, \lfloor c \rfloor$ resp. $\lfloor c \rfloor + 2, \ldots, N$ with probability $1/c$ resp. $1/(N - c)$ and to server $\lfloor c \rfloor + 1$ with probability $1 - \lfloor c \rfloor / c$ resp. $1 - (N - 1 - \lfloor c \rfloor)/(N - c)$. In particular, note that i) server $\lfloor c \rfloor + 1$ is the only server that can receive jobs that are predicted as short or long and this happens only if c is not an integer, ii) if $c < 1$ resp. $c > N - 1$, then no server that only processes jobs predicted as short resp. long exists. The impact of server $\lfloor c \rfloor + 1$ is negligible in the analysis developed in Section 3 where $N \to \infty$. Here, without loss of generality, one could restrict to the case where c is an integer. When N is finite, however, as in the analysis developed in Section 4, it is necessary to let c take non-integral values as otherwise it may not be possible to find a dispatching policy that guarantees stability.

The routing strategy described above is meant to reduce the interference between short and long jobs by mimicking the behavior of size-based routing policies discussed in Section 1.2. Here, the underlying principle is that a server only accepts jobs of size within a certain range so that short and long jobs are separated. Of course, within our framework it is not guaranteed that short and long jobs are separated from each other as in general exact sizes are not known a priori.

2.2 Testing Algorithm: Quality of Prediction vs Running Time

The quality of job-size predictions depends on the amount of processing time dedicated to the execution of the testing algorithm σ. In the literature, it is assumed that the exact size (or processing time) of each job is revealed once the testing algorithm is completed [1, 2, 19]. In contrast, we follow a more general approach where after testing the scheduler disposes of a probability distribution about the exact size that depends on σ. We define the connection between the quality of predictions and running time through

$$P_{x,y}(\sigma) := \mathbb{P}(X = x \cap Y_\sigma = y), \tag{3}$$

where $x, y \in \{x_m, x_M\}$ and $\sigma \geq 0$, and $P(\sigma) := [P_{x,y}(\sigma) : x, y \in \{x_m, x_M\}]$, i.e., the joint probability mass function of (X, Y_σ). In practice, this matrix can be learned from the data by user profiling methods. We do not impose a specific structure on $P(\sigma)$ but we will rely on the following assumption, which will be implicitly taken in the remainder of the paper.

ASSUMPTION 1. *The following properties hold:*

i) *$P(\sigma)$ is continuous in σ and, on a right-neighborhood of zero, differentiable;*
ii) *$P_{x,x}(\sigma)$ is non-decreasing and converges to $\mathbb{P}(X = x_m)$ as $\sigma \to \infty$, with $x \in \{x_m, x_M\}$;*
iii) *$P_{x,y}(\sigma)$ is non-increasing and converges to zero as $\sigma \to \infty$, with $x \neq y$ and $x, y \in \{x_m, x_M\}$.*

The rationale behind this assumption is as follows. The monotonicity properties in points ii) and iii) of Assumption 1 are natural because one expects that more information about the exact size is retrieved as the testing time increases. The asymptotic properties in the limit where $\sigma \to \infty$ are consistent with the idea that the testing algorithm is able to get the exact size of each job if it is executed for a sufficiently large amount of time. Finally, we claim that continuity is also natural while differentiability will be required for technical reasons and only used for Theorems 1 and 2.

REMARK 2. *In Assumption 1, we have defined $P(\sigma)$ on the whole non-negative real line for generality and because in principle the computational resources that are required by the testing algorithm are unbounded; recall that the "halting problem" is undecidable. However, we stress that we will consider "short" testing times $\sigma < \Lambda^{-1}$ as these rule out instability at the scheduler.*

For now, we do not impose any other assumption on the structure of Y_0, the random variable of job-size predictions when $\sigma = 0$: Y_0 may depend on X or not. Moreover, Y_σ and X are in general different in distribution. This models the fact that it may be faster to learn how to make good predictions about short rather than long jobs as the testing time increases.

2.3 Performance Measure

The performance measure of interest is the mean waiting time of jobs, i.e., the sum of the mean delay at the scheduler, i.e., (2), and the mean waiting time at the servers, say $R_c^{(N)}(\sigma)$. Within the given set of dispatching policies, the latter corresponds to the mean waiting time of three parallel M/G/1 queues because i) the output process of an M/M/1 queue, i.e., of the scheduler, is Poisson (by Burke's theorem) and ii) the thinning of a Poisson process produces again a Poisson process. Using the Pollaczek–Khinchine formula [6], we obtain

$$R_c^{(N)}(\sigma) = \frac{\Lambda}{2}\frac{\mathbb{P}(Y_\sigma = x_m)^2 p_m^2 \mathbb{E}[X^2 \mid Y_\sigma = x_m]}{\lfloor c \rfloor - \Lambda \mathbb{P}(Y_\sigma = x_m) p_m \, \mathbb{E}[X \mid Y_\sigma = x_m]} \mathbb{I}_{\{c \geq 1\}} + \frac{\Lambda}{2}\frac{p_z^2 \mathbb{E}[Z^2]}{1 - \Lambda p_z \, \mathbb{E}[Z]} + \frac{\Lambda}{2}\frac{\mathbb{P}(Y_\sigma = x_M)^2 p_M^2 \mathbb{E}[X^2 \mid Y_\sigma = x_M]}{N - 1 - \lfloor c \rfloor - \Lambda \mathbb{P}(Y_\sigma = x_M) p_M \, \mathbb{E}[X \mid Y_\sigma = x_M]} \mathbb{I}_{\{c \leq N-1\}} \tag{4}$$

provided that $\Lambda \mathbb{P}(Y_\sigma = x_m) p_m \, \mathbb{E}[X \mid Y_\sigma = x_m] < \lfloor c \rfloor$, $\Lambda p_z \, \mathbb{E}[Z] < 1$ and $\Lambda \mathbb{P}(Y_\sigma = x_M) p_M \, \mathbb{E}[X \mid Y_\sigma = x_M] < N - 1 - \lfloor c \rfloor$ and infinity otherwise. In (4),

$$p_m := \frac{\lfloor c \rfloor}{c} \mathbb{I}_{\{c \geq 1\}}, \quad p_M := \frac{N - 1 - \lfloor c \rfloor}{N - c} \mathbb{I}_{\{c \leq N-1\}}, \quad p_z := \mathbb{P}(Y_\sigma = x_m)\,(1 - p_m) + \mathbb{P}(Y_\sigma = x_M)\,(1 - p_M)$$

and Z is an auxiliary random variable equal to $X \mid Y_\sigma = x_m$ with probability $\mathbb{P}(Y_\sigma = x_m)(1 - p_m)/p_z$ and to $X \mid Y_\sigma = x_M$ otherwise. If $p_z > 0$, the three terms in (4) refer to the mean waiting time in the groups of servers $\{1, \ldots, \lfloor c \rfloor\}$, $\{\lfloor c \rfloor + 1\}$ and $\{\lfloor c \rfloor + 2, \ldots, N\}$, respectively. Therefore, the mean user-perceived waiting time is $\frac{\sigma}{1 - \Lambda\sigma} + R_c^{(N)}(\sigma)$. To make our approach a little bit more general and to better show the impact of the scheduler, we take as the performance measure of interest the quantity

$$D_c^{(N)}(\sigma) := f\left(\frac{\sigma}{1 - \Lambda\sigma}\right) + R_c^{(N)}(\sigma) \tag{5}$$

where $f : \mathbb{R}_+ \to \mathbb{R}_+$ is any increasing, continuous and differentiable function such that $f(0) = 0$.

We will also be interested in the *efficiency* of job-size testing relative to the case where no testing is performed ($\sigma = 0$), which we define by

$$\mathcal{E}(\sigma) := \mathcal{E}^{(N)}(\sigma) := \frac{\min_{c \in [0,N]} D_c^{(N)}(\sigma)}{\min_{c \in [0,N]} D_c^{(N)}(0)}. \tag{6}$$

Figure 2 shows plots of $\mathcal{E}(\sigma)$ with respect to a number of scenarios; see the figure caption for details. The main purpose here is to have a glimpse of the behavior of $\mathcal{E}(\sigma)$ and to highlight that the conditions on σ under which job-size testing improves performance, i.e., $\mathcal{E}(\sigma) < 1$, are non-trivial. For instance, when $N = 100$, $\rho = 0.8$ and $x_M = 10^5$, we also note that job-size testing may never improve if $\beta \in \{1.5, 2\}$ while bringing enormous benefits if in the same conditions β moves to 0.5 (increased job-size variability).

2.4 Additional Notation

Given a real function $g(t)$, we let $g'(t) := \frac{\mathrm{d}g(t)}{\mathrm{d}t}$ and if c is a constant, then $g'(c) := \frac{\mathrm{d}g(t)}{\mathrm{d}t}\Big|_{t=c}$. We let $\mathbb{I}_{\{A\}}$ denote the indicator function of the event A. If $\mathbb{I}_{\{A\}} = 0$, we take the convention that $\mathbb{I}_{\{A\}}/0 = \mathbb{I}_{\{A\}} \times \infty = 0$. Finally, := denotes mathematical definitions.

3 LARGE SYSTEMS

In this section, we focus on large systems and analyze job-size testing in the limit where $N \to \infty$. Several works in the literature of size-based routing have focused on this regime, e.g., [4, 9, 36], which is well justified because of the large number of servers that compose existing HPC systems. Within this regime, the testing time must converge to zero as otherwise the stability condition at the scheduler, i.e., $\lambda N \sigma < 1$, is not satisfied. Thus, to understand under which conditions job size testing is effective in reducing waiting times with respect to the situation where jobs run

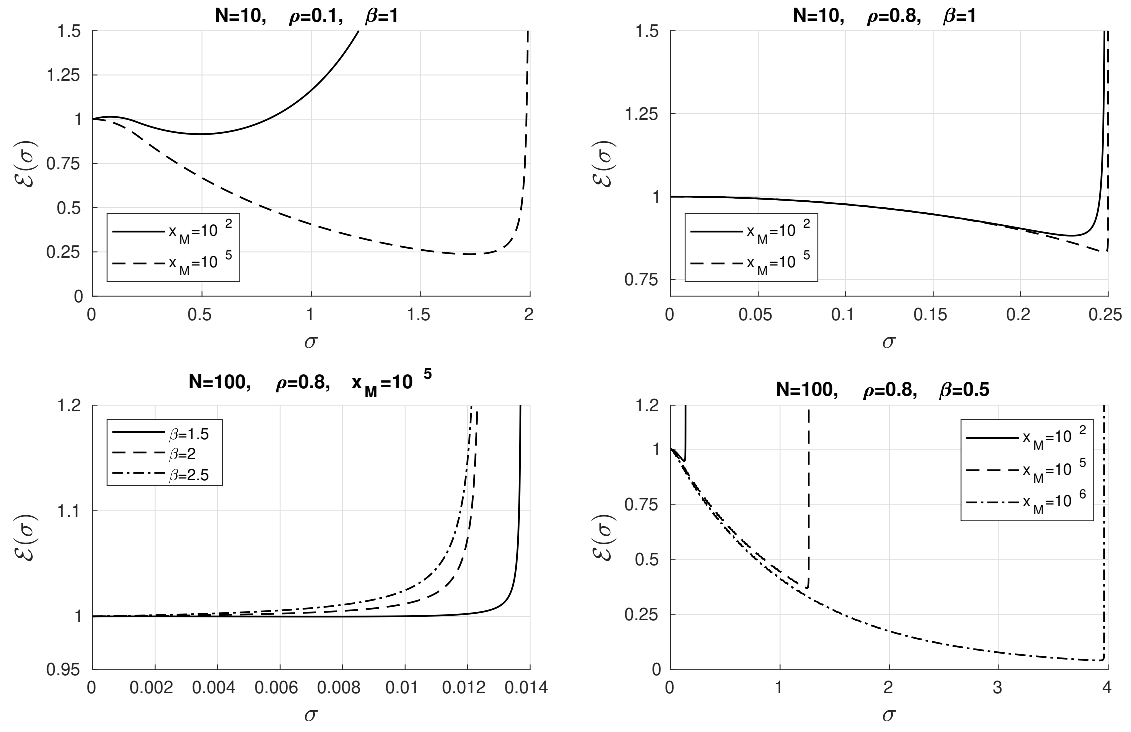

Fig. 2. Plots of the efficiency measure $\mathcal{E}$, see (6), assuming that (1) holds with $\alpha = 1$ and $f(\cdot) = \cdot$. Also, $P_{x_m,x_m}(\sigma) = (1 - e^{-10\sigma})(\mathbb{P}(X = x_m) - P_{x_m,x_m}(0)) + P_{x_m,x_m}(0)$, $P_{x_M,x_M}(\sigma) = (1 - e^{-\sigma})(\mathbb{P}(X = x_M) - P_{x_M,x_M}(0)) + P_{x_M,x_M}(0)$, $P_{x_m,x_M}(\sigma) = \mathbb{P}(X = x_m) - P_{x_m,x_m}(\sigma)$ and $P_{x_M,x_m}(\sigma) = \mathbb{P}(X = x_M) - P_{x_M,x_M}(\sigma)$, which means that the profile matrix $P_a(\sigma)$ agrees with the law of diminishing returns.

untested, we investigate when $D_c^{(N)}(\sigma)$ is decreasing on a right neighborhood of zero for all N sufficiently large. More specifically, given a sequence of cutoff servers $(C_N)_N$, we study the limit as $N \to \infty$ of the derivative of $D_{C_N}^{(N)}(\sigma)$ on 0^+.

We now specify the limiting regime of interest in detail and then perform the resulting analysis under two complementary assumptions on Y_0. To quickly summarize the results in this section, our findings show that job size testing is counterproductive in large systems unless the system is in heavy-traffic and short jobs can be predicted reliably.

3.1 Limiting Regime

Let

$$C_N := c^* N + h_N \tag{7}$$

where

$$c^* := (1 - \lambda \mathbb{E}[X \mid Y_0]) \frac{\sqrt{\mathbb{E}[X^2 \mid Y_0 = x_m]}}{\mathbb{E}[\sqrt{\mathbb{E}[X^2 \mid Y_0]}]} \mathbb{P}(Y_0 = x_m) + \lambda \, \mathbb{E}[X \mid Y_0 = x_m] \mathbb{P}(Y_0 = x_m) \tag{8}$$

and $(h_N)_N$ is a uniformly bounded sequence. The intuition is that c^*N represents the cutoff that minimizes the mean response time of the system with N servers (see the proof of Proposition 1 in the Appendix); note that this is not necessarily an integer and this explains why we have added the sequence h_N. One can check that if the cutoff server is C_N, i.e., $c = C_N$, then the system with N servers is stable for all N large enough.

The following proposition provides structural properties on the *asymptotic optimal mean waiting time*, say R^*; see the Appendix for a proof. For any $\tau \geq 0$, it assumes that σ is designed such that

$$\sigma = \sigma^{(N)}(\tau) = \frac{1}{\lambda N + \tau^{-1}\, o(N)} \tag{9}$$

if $\tau > 0$ and $\sigma = 0$ if $\tau = 0$; clearly, the $o(N)$ term in (9) does not depend on τ. This design choice for the testing time is convenient because it satisfies the stability condition at the scheduler while imposing a controlled budget on its performance cost. In fact, the mean delay at the scheduler (2) boils down to $\tau/o(N)$.

Proposition 1. *Assume that* (9) *holds. For any* $\tau \geq 0$,

$$R^* := \lim_{N\to\infty} \min_{c\in[0,N]} R_c^{(N)}(\sigma) = \lim_{N\to\infty} R_{C_N}^{(N)}(\sigma) = \frac{\lambda}{2} \frac{\left(\mathbb{E}\left[\sqrt{\mathbb{E}[X^2 \mid Y_0]}\right]\right)^2}{1-\lambda\mathbb{E}[X]}. \tag{10}$$

The limit R^* is made explicit and the conditional expectation term in (10) reveals the impact of predictions on the optimal system performance.

Remark 3. *Any sequence of cutoff servers that satisfies* (7) *induces the optimal performance at the servers. This motivates us to study the model under investigation with respect to a sequence of systems indexed by* N *where for the* N*-th system the cutoff server is* C_N.

3.2 Independent Predictions for Zero Testing Time

When $\sigma = 0$, no testing algorithm is executed and in this case, it is reasonable to assume that no additional information about the true job size is revealed at the moment of its arrival. Since the scheduler knows the distribution of X, its best strategy to predict job sizes consists in sampling independently from such distribution. This leads us to consider the case where X and Y_0 are independent and identically distributed, for which we have the following negative result; see the Appendix for a proof.

Theorem 1. *Assume that* X *and* Y_0 *are independent and equal in distribution. Then,*

$$\lim_{N\to\infty} \lim_{\sigma\downarrow 0} \frac{\mathrm{d}}{\mathrm{d}\sigma} D_{C_N}^{(N)}(\sigma) = f'(0). \tag{11}$$

Note that the LHS term of (11) is interpreted as the change in performance when a small testing time is applied to a large system. Since $f'(0) > 0$, $\sigma \mapsto D_{C_N}^{(N)}(\sigma)$ is strictly increasing on a right neighborhood of zero, for all N large enough. Thus, it turns out that job testing does not help in this case. While $R_{C_N}^{(N)}(\sigma)$ decreases on a right neighborhood of zero for any (finite) N, in the proof of Theorem 1 we show that $\lim_{N\to\infty} \lim_{\sigma\downarrow 0} \frac{\mathrm{d}}{\mathrm{d}\sigma} R_{C_N}^{(N)}(\sigma) = 0$, which shows that the extra cost of job testing does not eventually pay off in the large system limit.

3.3 No False Small

Now, we analyze the effectiveness of the proposed job testing strategy under some additional practical assumptions on the structure of predictions. Specifically, we are interested in a scenario where predictions about *short* jobs are precise and obtained instantaneously, while predictions of long jobs are subject to errors. This setting is considered in, e.g., [38, Table 6]. It is justified because just the (little) knowledge of the fact that the job is executed against "small" parameters, which in practice can be easily obtained almost instantaneously, often implies that the job itself is short, while if it is predicted as long, it may be long or short. For instance, in types of jobs related to i) linear algebra (matrix inversions, etc.), ii) machine learning, which train a machine learning model by running a suitable number of "iterations", iii) image processing, it should be clear that if i) the matrix dimension, ii) the number of iterations, iii) the number of pixels is small, then the job is short. In contrast, if these parameters are large, then the job may not be necessarily long because the matrix may be sparse or because iteration steps may exploit data locality. Within this scenario,

$$Y_\sigma = x_m \text{ implies } X = x_m$$

while $X \in [x_m, x_M]$ within the event $Y_\sigma = x_M$. This leads us to consider a setting where

$$P_{x_M,x_m}(\sigma) = 0, \quad P_{x_m,x_m}(\sigma) > 0 \tag{12}$$

for all $\sigma \geq 0$. Inspired from common parlance in hypothesis testing, we refer to this scenario as "No False Small".

Using the definition of conditional probability and given that the distribution of X is fixed, (12) implies that the following conditions hold true for all $\sigma \geq 0$:

$$P_{x_M,x_M}(\sigma) = \mathbb{P}(X = x_M) \tag{13a}$$

$$\mathbb{P}(X = x_m \mid Y_\sigma = x_M) = \frac{\mathbb{P}(X = x_m)}{\mathbb{P}(Y_\sigma = x_M)} \tag{13b}$$

$$\mathbb{P}(Y_\sigma = x_m) = P_{x_m,x_m}(\sigma) \leq \mathbb{P}(X = x_m). \tag{13c}$$

Thus, job-size predictions are larger than their exact counterparts. This is the desired scenario because underprediction is technically unacceptable in practice as discussed in [33]. Also, note that this scenario is different from the one considered in Theorem 1 because (12) rules out the case where X and Y_0 are independent. This may be interpreted as follows: when a job arrives, it immediately reveals some of its features, e.g., its parameters, which provide some information about its size.

Within this setting, the following result provides the conditions under which testing improves performance.

Theorem 2. *Let (12) hold. Then,*

$$\lim_{N\to\infty} \lim_{\sigma\downarrow 0} \frac{\mathrm{d}}{\mathrm{d}\sigma} D_{C_N}^{(N)}(\sigma) = f'(0) - P'_{x_m,x_m}(0) \frac{\lambda \mathbb{E}[\sqrt{\mathbb{E}[X^2|Y_0]}]}{1 - \lambda \mathbb{E}[X]} \left(\frac{\mathbb{E}[X^2 \mid Y_0 = x_M] + x_m^2}{2\sqrt{\mathbb{E}[X^2 \mid Y_0 = x_M]}} - x_m \right). \tag{14}$$

In particular, for all N large enough, $\sigma \mapsto D_{C_N}^{(N)}(\sigma)$ is decreasing on a right neighborhood of zero if

$$P'_{x_m,x_m}(0) \frac{\lambda}{2} \frac{(\sqrt{\mathbb{E}[X^2]} - x_m)^2}{1 - \lambda \mathbb{E}[X]} \geq f'(0). \tag{15}$$

Thus, job-size testing is effective in heavy traffic ($\lambda \mathbb{E}[X] \uparrow 1$) or if job sizes are highly variable, as in this case $\sqrt{\mathbb{E}[X^2]} \gg x_m$. In contrast, it is ineffective in light traffic ($\lambda \downarrow 0$) or if $P_{x_m,x_m}(0)$ is close to its asymptotic limit $\mathbb{P}(X = x_m)$. In the latter case, we can expect $P'(0) \approx 0$ because $P(t)$ is non-decreasing, and therefore (14) is not satisfied.

4 HEAVY TAILED DISTRIBUTIONS

Complementary to the approach in the previous section, we now keep the system size N constant and analyze job testing in a different limiting regime where the job size variability grows large. Towards this purpose, we investigate under which choices of σ it is possible to make the efficiency measure $\mathcal{E}(\sigma)$ (see (6)) less than one or arbitrarily small while keeping N constant.

The approach considered here is orthogonal to the one followed in Section 3. To quickly summarize the results in this section, our findings show that job size testing is *very* effective in presence of heavy-tailed job size distributions provided that testing times are properly chosen.

4.1 Limiting Regime

We consider a limiting regime where X satisfies (1) with $0 < \beta < 1$ and $x_M \to \infty$. This regime limits the investigation to job size distributions where the first two moments and the coefficient of variation of X grow to infinity.

Several works in the literature have investigated regimes where the coefficient of variation of X grows to infinity, e.g., [20, 27, 32]. In fact, it is well-known that the introduction of size-based routing and related techniques were motivated by the observation that job sizes with heavy-tailed distributions are common in empirical studies of computer systems [14, 22, 26].

4.2 Efficiency of Job-Size Testing

Let

$$R^{\downarrow} := \frac{\lambda}{2} \frac{\mathbb{E}[X]^2}{1 - \lambda \mathbb{E}[X]}. \tag{16}$$

The next proposition, proven in the appendix, shows that $R^{\downarrow}$ is a lower bound on the mean waiting time at the servers. This fact does not play a role in the proof of Theorem 3 but will ensure that the response time at the scheduler is smaller than the mean delay at the scheduler, which may be desired from a practical point of view. The key property on the structure of $R^{\downarrow}$ will be that it is of the order of $\mathbb{E}[X]$, provided that the network load $\lambda \mathbb{E}[X]$ is constant.

Proposition 2. $R_c^{(N)}(\sigma) \geq R^{\downarrow}$.

A straightforward consequence of Proposition 2 is the following lower bound on $\mathcal{E}(\sigma)$.

Corollary 1. *For all* σ, $\mathcal{E}(\sigma) \geq \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}$.

If $\beta > 2$, i.e., if X is not variable enough, then $\mathbb{E}[X]^2/\mathbb{E}[X^2] \to 1$ as $x_M \to \infty$, and thus job-size testing is clearly inefficient.

Now, we consider the more interesting case where job sizes have increased variability. The idea here is to use $R^{\downarrow}$ as an handle on the response time at the scheduler to accordingly design a testing time able to make the efficiency measure $\mathcal{E}(\sigma)$ arbitrarily small. Assume that σ is designed such that $\frac{\sigma}{1-\Lambda\sigma} = \frac{R^{\downarrow}}{\gamma}$, for some constant $\gamma > 0$. In the following, let us denote such testing time by σ^*. In this case, the following proposition shows that $D_c^{(N)}$ scales at most as $R^{\downarrow}$ provided that x_M and N are sufficiently large; see the Appendix for a proof.

Proposition 3. *Let* $\rho \in (0, 1)$ *and* $\gamma > 0$ *be constants independent of* x_M. *Let also*

$$\sigma^* = \left(\lambda N + \gamma / R^{\downarrow}\right)^{-1} \tag{17}$$

and

$$c := \begin{cases} \max\{Nc^\star, 1\}, & \text{if } N(1-\rho) > 1 \\ \max\{Nc^\star, N(1-\rho) - N\rho\,\phi(x_M)\}, & \text{otherwise} \end{cases} \tag{18}$$

where $\phi(x_M)$ is such that $\lim_{x_M\to\infty} \phi(x_M)x_M^\theta = 1$, for some $\theta \in (0,\beta)$, and

$$c^\star := (1-\rho)\frac{\sqrt{\mathbb{E}[X^2 \mid Y_{\sigma^\star} = x_m]}}{\mathbb{E}[\sqrt{\mathbb{E}[X^2 \mid Y_{\sigma^\star}]}]}\mathbb{P}(Y_{\sigma^\star} = x_m) + \lambda\,\mathbb{E}[X \mid Y_{\sigma^\star} = x_m]\mathbb{P}(Y_{\sigma^\star} = x_m).$$

Assume that

i) $P_{x_M,x_m}(t) = O(t^{-\eta_m})$ *with* $\eta_m > \frac{2}{1-\beta}$ *and* $P_{x_m,x_M}(t) = O(t^{-\eta_M})$ *with* $\eta_M > \frac{\beta}{1-\beta}$

ii) *the job size distribution satisfies* (1) *with* $\beta < 1$*;*

iii) $\lambda\mathbb{E}[X] = \rho$.

Then, if $N(1-\rho) > 1$

$$\lim_{x_M\to\infty} \frac{D_c^{(N)}(\sigma^*)}{R^\downarrow} = \frac{1}{\gamma} + \frac{N(1-\rho)}{N(1-\rho)-1}, \tag{19}$$

otherwise

$$\lim_{x_M\to\infty} \frac{D_c^{(N)}(\sigma^*)}{x_M^\theta R^\downarrow} = (1-\rho)\frac{N-1}{N\rho^2}. \tag{20}$$

Now, to investigate the efficiency ratio $\mathcal{E}(\sigma)$, it is enough to compare $\min_{c\in[0,N]} D_c^{(N)}(0)$ with respect to $R^\downarrow$, which is easier. Before doing that, let us elaborate on the assumptions in Proposition 3:

- The expressions in (17) and (18) are design choices and serve to have a handle on the mean delay at the scheduler and at the servers. Note that (2) boils down to $R^\downarrow/\gamma$, which is smaller than $R_c^{(N)}(\sigma^*)$ (if $\gamma \geq 1$) in view of Proposition 2, and that the expression of c in (18) somewhat minimizes $c \mapsto R_c^{(N)}(\sigma^*)$.
- Item i) is technical and states that the probability of having a long resp. short but predicted-short resp. predicted-long job decays sufficiently fast with the testing time; recall that $P_{x,y}(t) \to 0$ as $t \to \infty$ by Assumption 1, provided that $x \neq y$. In the "No False Small" scenario discussed in Section 3.3, this condition is clearly satisfied.
- Item ii) limits the investigation to job size distributions with a large coefficient of variation, which is in agreement with what is observed in practice [27].
- Item iii) keeps the network load at the desired level ρ.

Within the setting of Proposition 3, we can show that the efficiency ratio at the testing time σ^* can be arbitrarily small. This is stated precisely in the following result.

Theorem 3. *Let the assumptions of Proposition 3 hold and suppose also that one of the following conditions holds true:*

i) X *and* Y_0 *are independent,*

ii) (12) *holds and* $P_{x_m,x_m}(0) \leq 1 - O(x_M^{-\theta})$*, for any* $\theta \in (0,\beta)$.

Then,

$$\lim_{x_M\to\infty} \mathcal{E}(\sigma^*) = 0. \tag{21}$$

Theorem 3 analyzes the efficiency index (6) in the scenarios discussed in Sections 3.2 and 3.3. In the "No False Small" scenario, i.e., case ii), the desired property (21) holds under the additional condition that $P_{x_m,x_m}(0)$ is sufficiently small.

This is somewhat necessary: if $P_{x_m,x_m}(0)$ would be too close to one, which can be expected because $P_{x_m,x_m}(0)$ can be arbitrarily close to $\mathbb{P}(X = x_m)$ and $\mathbb{P}(X = x_m) = 1 - \alpha x_M^{-\beta} \approx 1$ if x_M is large, then there would be no margin for job testing to further gather useful information on job sizes. In this case, the extra cost induced by σ^* does not pay off.

If X and Y_0 are independent and $N(1-\rho) > 1$, in the proof of Theorem 3 we actually show that $\lim_{x_M \to \infty} \mathcal{E}(\sigma^*) x_M^{\theta} = 0$ for any $\theta \in (0, \beta)$, which is slightly stronger than (21).

5 NUMERICAL RESULTS

We now investigate the efficiency ratio $\mathcal{E}$ numerically with respect to realistic job size distributions from neuroscience workflows drawn from [23]. Here, jobs are divided into two categories, short and long, with the mean execution time of small jobs of around 25 minutes (i.e., $x_m = 25$) and that of large jobs of around 9 hours (i.e., $x_M = 540$).

5.1 Efficiency Assessment

We start our numerical analysis distinguishing three cases:

(1) equal number of small and large jobs ($\mathbb{P}(X = x_m) = 0.5$, $\mathbb{E}[X] = 282.5$),
(2) workload with 80% large jobs ($\mathbb{P}(X = x_m) = 0.2$, $\mathbb{E}[X] = 437$), and
(3) workload with 20% large jobs ($\mathbb{P}(X = x_m) = 0.8$, $\mathbb{E}[X] = 128$).

As observed on the ACCRE cluster (the Vanderbilt high-performance computing cluster), job submissions follow a Poisson process with mean inter-arrival times of 8 minutes [23], which we assume in the remainder.

To completely parameterize our model, it remains to choose a profile matrix $\sigma \mapsto P_\sigma$ as a function of the testing time σ. Here, we distinguish the "Independent Predictions for Zero Testing Time" scenario discussed in Section 3.2 and the "No False Small" scenario discussed in Section 3.3. For the former, we assume that

$$\begin{aligned}
P_{x_m,x_m}(\sigma) &= (1 - e^{-a\sigma})(p - P_{x_m,x_m}(0)) + P_{x_m,x_m}(0) \\
P_{x_M,x_M}(\sigma) &= (1 - e^{-b\sigma})(1 - p - P_{x_M,x_M}(0)) + P_{x_M,x_M}(0) \\
P_{x_m,x_M}(\sigma) &= p - P_{x_m,x_m}(\sigma) \\
P_{x_M,x_m}(\sigma) &= 1 - p - P_{x_M,x_M}(\sigma)
\end{aligned}$$

and for the latter, we assume that

$$\begin{aligned}
P_{x_m,x_m}(\sigma) &= (1 - e^{-a\sigma})(p - P_{x_m,x_m}(0)) + P_{x_m,x_m}(0) \\
P_{x_M,x_M}(\sigma) &= 1 - p \\
P_{x_m,x_M}(\sigma) &= p - P_{x_m,x_m}(\sigma) \\
P_{x_M,x_m}(\sigma) &= 0
\end{aligned} \tag{22}$$

where $p := \mathbb{P}(X = x_m)$. These structures make sense from a practical point of view because they agree with the law of diminishing returns. In addition, they satisfy Assumption 1 and the assumption in Theorem 3. The parameters a and b are interpreted as how fast the predictions of short and long jobs become accurate, respectively. It should be intuitive that in practice it is easier to estimate short rather than long jobs. Thus, in our parameter setting, we have used $b = 1$ and $a = 3$; within higher values of a, e.g., $a = 10$, we have noticed no essential change in the numerical results that follow.

Within this setting, Figure 3 plots the resulting efficiency measure $\mathcal{E}$, see (6), for different values of the network load ρ and system size N. The vertical lines denote the heuristic testing time choice given in (17) used in Theorem 3, where we have set $\gamma = 10$ for all plots. The blue resp. red lines refer to a system with $N = 10$ resp. $N = 100$ servers In all cases, it turns out that it is possible to find a testing time that makes $\mathcal{E}$ strictly smaller than one and our heuristic testing time choice σ^* provides a near-optimal performance in all cases. In addition, $\mathcal{E}$ is always decreasing on a right neighborhood of zero, which means that even a cheap job inspection pays off to reduce the mean waiting time. From the figure, we also notice that the gains are more pronounced when there are many small jobs and fewer long jobs (the plots on the leftmost column), which is the most frequent case in practice.

5.2 Beyond Two-point Distributions

We now consider a setting where job sizes follow a more general distribution. The goal is to numerically show that the two-point approximation assumption used in our model is robust and thus that the insights identified above hold true even in more general settings. Towards this purpose, we assume that X follows the bimodal distribution considered in [23, Figure 6.d]. Specifically, X follows a truncated normal distribution with mean $x_m = 25$ resp. $x_M = 540$ minutes and standard deviation $\sigma_m = 10$ resp. $\sigma_M = 200$ with probability 0.8 resp. 0.2. To better stress the robustness of our model, the standard deviations used here are much larger than the ones considered in [23, Figure 6.d]. Truncation is used to ensure that job sizes are non-negative. Letting $f_X(x)$ denote the density function of X, we have

$$f_X(x) = 0.8\frac{\Phi'_{x_m,\sigma_m}(x)}{1-\Phi_{x_m,\sigma_m}(0)} + 0.2\frac{\Phi'_{x_M,\sigma_M}(x)}{1-\Phi_{x_M,\sigma_M}(0)} \tag{23}$$

for all $x \geq 0$ and $f_X(x) = 0$ otherwise, where $\Phi_{a,b}(x)$ denotes the cumulative distribution function of the normal distribution with mean a and standard deviation b at point x. Using the empirical rule of the normal distribution, we say by design that a job is short if its size is smaller than $\bar{x} := x_m + 5\sigma_m$, and long otherwise. Figure 4 illustrates a plot of the density function of X.

To apply the proposed framework, we need to extend the structure of the profile matrix P in (3). For any $\sigma \geq 0$, let $f_{X,Y_\sigma}(x,y)$ denote the joint density of the random pair (X, Y_σ) where $x \in \mathbb{R}_+$, $y \in \{x_m, x_M\}$. Here, we interpret the event $Y_\sigma = x_m$ resp. $Y_\sigma = x_M$ as the job is predicted as short resp. long. Now, due to space constraints, we consider the "No False Small" scenario only, and we extend the conditions in (22) to get

$$\begin{aligned}
f_{X,Y_\sigma}(x,x_m) &= (1-e^{-a\sigma})(f_X(x) - f_{X,Y_\sigma}(0,x_m)) + f_{X,Y_\sigma}(0,x_m), \quad \forall x < \bar{x}\\
f_{X,Y_\sigma}(x,x_M) &= f_X(x), \quad \forall x \geq \bar{x}\\
f_{X,Y_\sigma}(x,x_M) &= f_X(x) - f_{X,Y_\sigma}(x,x_m), \quad \forall x < \bar{x}\\
f_{X,Y_\sigma}(x,x_m) &= 0, \quad \forall x \geq \bar{x}.
\end{aligned}$$

Within this setting, we compare the efficiency measure $\mathcal{E}$, see (6), with the efficiency measure that would be obtained if X would be replaced by $\tilde{X}$ where $\tilde{X}$ is distributed on only two points. Specifically, $\tilde{X} = x_m = 25$ with probability 0.8 and $\tilde{X} = x_M = 540$ otherwise. As done above, we consider a system with 10 and 100 servers, loads 0.4 and 0.8 and fix $a = 10$. Figure 5 shows that i) both efficiency measures are very close to each other and ii) our heuristic choice for the testing times, which only depends on $\tilde{X}$, provides near-optimal performance even when the actual job size distribution satisfies (23). At least in the case of neuroscience applications or bimodal distributions, this also justifies our two-point approximation and numerically validates the claims described in Section 2.1.1.

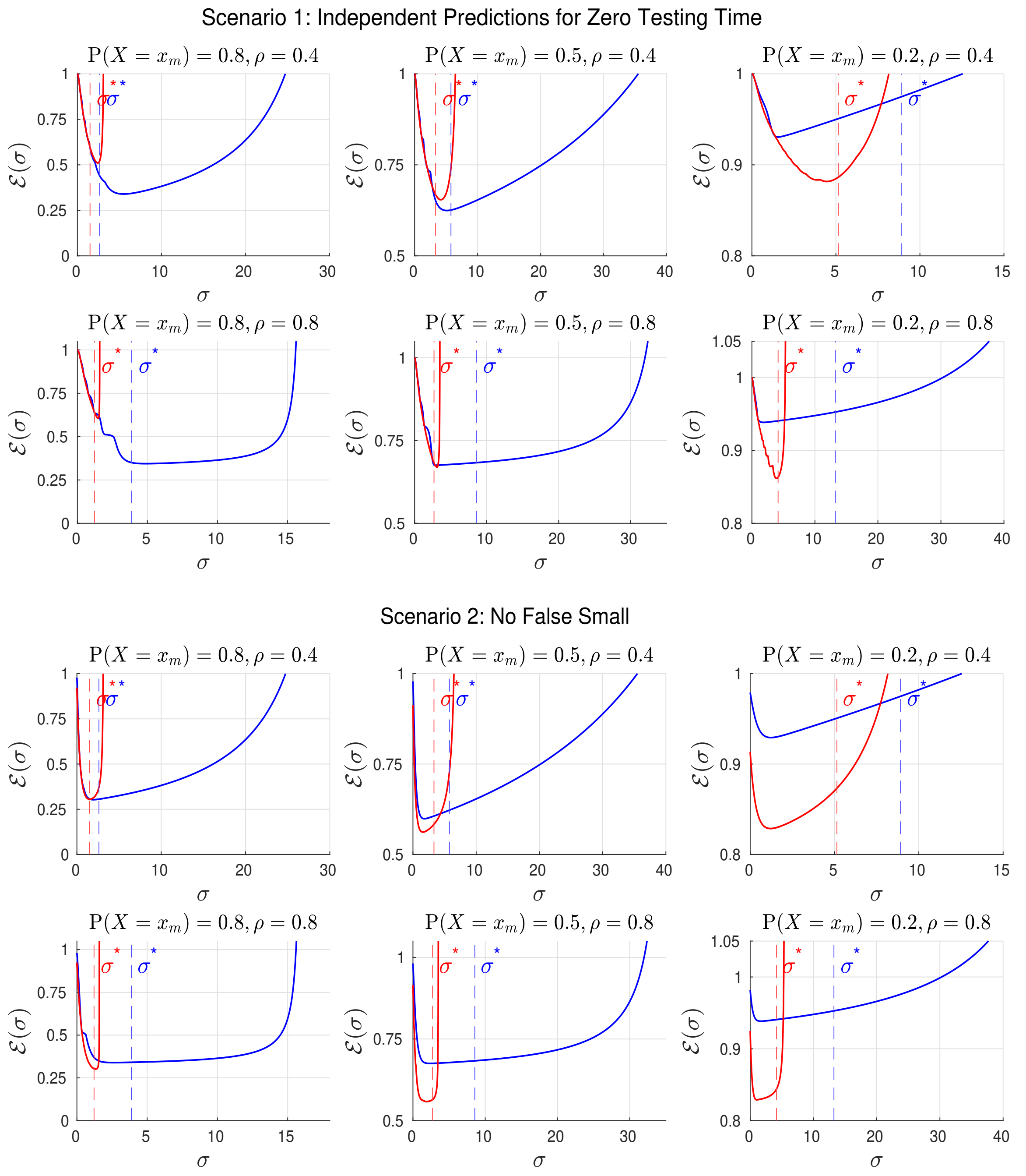

Fig. 3. Plots of the efficiency measure (6) with respect to job-size distributions from neuroscience applications [23], where $x_m = 25$ and $x_M = 540$. The vertical lines denote the heuristic testing time choice given in (17). Blue resp. red lines refer to a system with $N = 10$ resp. $N = 100$ servers.

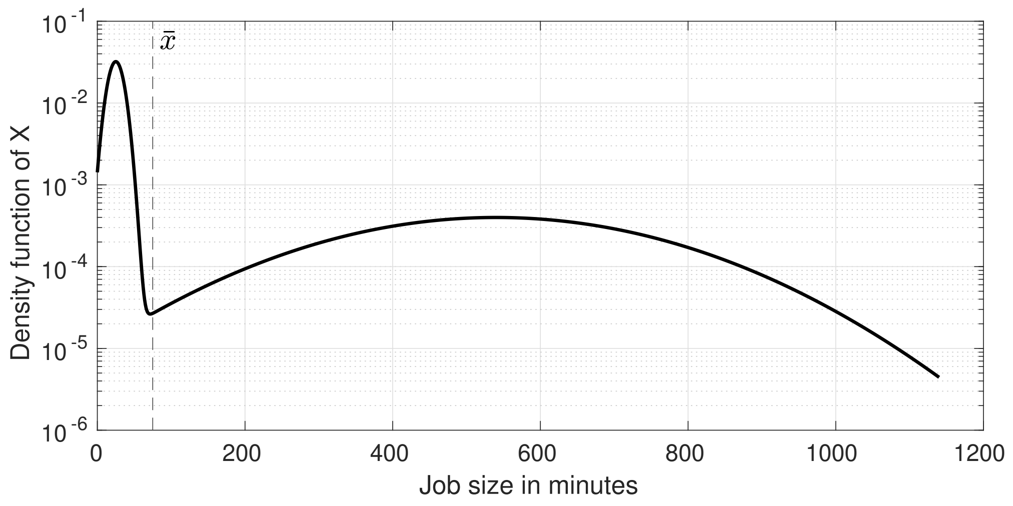

Fig. 4. Bimodal distribution of job sizes from neuroscience applications. The vertical dashed line distinguishes between short and long jobs.

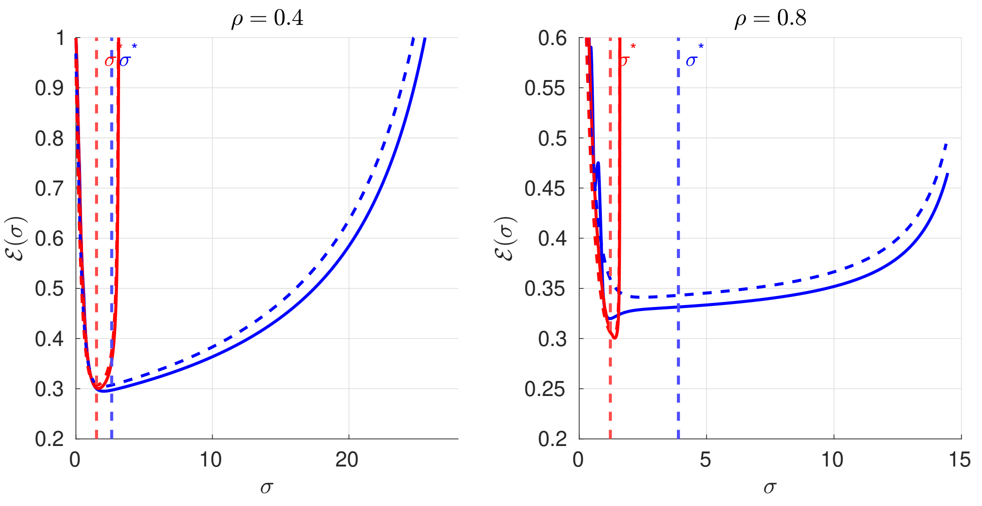

Fig. 5. Plots of the efficiency measure $\mathcal{E}$. The continuous resp. dashed lines refer to a system where job sizes are captured by X, with density given in (23), resp. $\tilde{X}$. Blue resp. red lines refer to a system with $N = 10$ resp. $N = 100$ servers. The vertical lines denote the heuristic testing time choice given in (17).

6 CONCLUSION AND PERSPECTIVES

We have proposed and analyzed a performance model for load balancing with job-size testing. The proposed model generalizes existing approaches to scheduling with testing, which only consider static settings with a finite number of jobs. Our main conclusion is that whether job-size testing brings a performance degradation or improvement strongly depends on the traffic conditions, system size and the coefficient of variation of job sizes. We have made explicit such conditions and these may help the system manager to design efficient testing policies that possibly adapt to changes in environmental conditions.

Let us outline some future perspectives from this work. First, a natural generalization of the proposed framework considers the case where job sizes follow a general probability distribution function. In this case, the joint probability mass function of (X, Y_σ) may be captured by $P = [P_{x,y}(\sigma), x \in \mathbb{R}_+, y \in \{x_m, x_M\}]$ where $P_{x,y}(\sigma) := \mathbb{P}(X \le x \cap Y_\sigma = y)$ and one can check that the expressions of $D_c^{(N)}(\sigma)$ and $\mathcal{E}(\sigma)$ given in (5) and (6) respectively are still valid as is. In addition, Propositions 1 and 2, together with Corollary 1, are also valid as their proofs do not use the particular structure of two-point distributions.

A further generalization considers a multiclass setting where jobs come from different sources. One may interpret jobs from a given class as jobs issued by the same user or application. Job size distributions are different across classes, though job sizes from a given source take at most two values (short or long). In this respect, one could argue that the size of a *random* job is general enough, provided that the number of classes is large enough. Here, one may also consider class-dependent testing times rather than one for all classes. We believe that many of the insights found in this paper hold even within such generalization, though a deeper investigation is left as future work.

Finally, our work assumes that the job size distribution is fixed. Though this is a common assumption in queueing theory, it is interesting to study the case where it changes over time sufficiently fast to prevent the average system performance to be captured by the stationary behavior of the underlying Markov process as we do. The analysis of this case would require a radically different approach because this prevents one to use the Pollaczek–Khinchine formula. However, the benefits of job size testing may be evaluated by identifying a reinforcement learning algorithm and investigating its *regret* over a finite horizon.

REFERENCES

[1] Susanne Albers and Alexander Eckl. 2020. Explorable Uncertainty in Scheduling with Non-Uniform Testing Times. In *Approximation and Online Algorithms: 18th International Workshop, WAOA 2020, Virtual Event, September 9–10, 2020, Revised Selected Papers*. Springer-Verlag, Berlin, Heidelberg, 127–142. https://doi.org/10.1007/978-3-030-80879-2_9

[2] Susanne Albers and Alexander Eckl. 2021. Scheduling with Testing on Multiple Identical Parallel Machines. In *Algorithms and Data Structures: 17th International Symposium, WADS 2021, August 9–11, 2021, Proceedings*. Springer-Verlag, Berlin, Heidelberg, 29–42. https://doi.org/10.1007/978-3-030-83508-8_3

[3] Saed Alizamir, Francis de Véricourt, and Peng Sun. 2013. Diagnostic Accuracy Under Congestion. *Management Science* 59, 1 (2013), 157–171. https://doi.org/10.1287/mnsc.1120.1576

[4] Jonatha Anselmi. 2020. Combining Size-Based Load Balancing with Round-Robin for Scalable Low Latency. *IEEE Transactions on Parallel and Distributed Systems* 31, 4 (2020), 886–896. https://doi.org/10.1109/TPDS.2019.2950621

[5] J. Anselmi and J. Doncel. 2019. Asymptotically Optimal Size-Interval Task Assignments. *IEEE Transactions on Parallel and Distributed Systems* 30, 11 (nov 2019), 2422–2433. https://doi.org/10.1109/TPDS.2019.2920121

[6] S. Asmussen and S. Asmussen. 2003. *Applied Probability and Queues*. Springer.

[7] Eitan Bachmat and Josu Doncel. 2021. Size-Based Routing Policies: Non-Asymptotic Analysis and Design of Decentralized Systems. *Sensors* 21, 8 (2021). https://www.mdpi.com/1424-8220/21/8/2701

[8] Eitan Bachmat, Josu Doncel, and Hagit Sarfati. 2019. Performance and Stability Analysis of the Task Assignment Based on Guessing Size Routing Policy. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 1–13. https://doi.org/10.1109/MASCOTS.2019.00012

[9] Eitan Bachmat and Hagit Sarfati. 2010. Analysis of SITA Policies. *Perform. Eval.* 67, 2 (Feb. 2010), 102–120.

[10] Cynthia Bailey Lee, Yael Schwartzman, Jennifer Hardy, and Allan Snavely. 2005. Are User Runtime Estimates Inherently Inaccurate?. In *Job Scheduling Strategies for Parallel Processing*, Dror G. Feitelson, Larry Rudolph, and Uwe Schwiegelshohn (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 253–263.

[11] Luis Diego Briceno, Bhavesh Khemka, Howard Jay Siegel, Anthony A. Maciejewski, Christopher Groër, Gregory Koenig, Gene Okonski, and Steve Poole. 2011. Time Utility Functions for Modeling and Evaluating Resource Allocations in a Heterogeneous Computing System. In *IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*. 7–19. https://doi.org/10.1109/IPDPS.2011.123

[12] João M.P. Cardoso, José Gabriel F. Coutinho, and Pedro C. Diniz. 2017. Chapter 5 - Source code transformations and optimizations. In *Embedded Computing for High Performance*, João M.P. Cardoso, José Gabriel F. Coutinho, and Pedro C. Diniz (Eds.). Morgan Kaufmann, Boston, 137–183.

[13] Wenyan Chen, Kejiang Ye, Yang Wang, Guoyao Xu, and Cheng-Zhong Xu. 2018. How Does the Workload Look Like in Production Cloud? Analysis and Clustering of Workloads on Alibaba Cluster Trace. In *2018 IEEE 24th Int. Conf. on Parallel and Distributed Systems (ICPADS)*. 102–109. https://doi.org/10.1109/PADSW.2018.8644579

[14] Mark E. Crovella, Murad S. Taqqu, and Azer Bestavros. 1998. A Practical Guide to Heavy Tails. Birkhauser Boston Inc., Cambridge, MA, USA, Chapter Heavy-tailed Probability Distributions in the World Wide Web, 3–25.

[15] Mark Van der Boor, Sem C. Borst, Johan S. H. Van Leeuwaarden, and Debankur Mukherjee. 2022. Scalable Load Balancing in Networked Systems: A Survey of Recent Advances. *SIAM Rev.* 64, 3 (2022), 554–622. https://doi.org/10.1137/20M1323746

[16] Sheng Di, Derrick Kondo, and Walfredo Cirne. 2012. Characterization and Comparison of Cloud versus Grid Workloads. In *2012 IEEE International Conference on Cluster Computing*. 230–238. https://doi.org/10.1109/CLUSTER.2012.35

[17] Fanny Dufossé, Christoph Dürr, Noël Nadal, Denis Trystram, and Óscar C. Vásquez. 2022. Scheduling with a processing time oracle. *Applied Mathematical Modelling* 104 (2022), 701–720. https://doi.org/10.1016/j.apm.2021.12.020

[18] Christoph Dürr, Thomas Erlebach, Nicole Megow, and Julie Meißner. 2018. Scheduling with Explorable Uncertainty. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 94)*, Anna R. Karlin (Ed.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 30:1–30:14. https://doi.org/10.4230/LIPIcs.ITCS.2018.30

[19] Christoph Dürr, Thomas Erlebach, Nicole Megow, and Julie Meißner. 2020. An Adversarial Model for Scheduling with Testing. *Algorithmica* 82, 12 (2020), 3630–3675.

[20] Muhammad El-Taha and Bacel Maddah. 2006. Allocation of Service Time in a Multiserver System. *Management Science* 52, 4 (2006), 623–637. https://doi.org/10.1287/mnsc.1050.0467

[21] Yuping Fan, Paul Rich, William E. Allcock, Michael E. Papka, and Zhiling Lan. 2017. Trade-Off Between Prediction Accuracy and Underestimation Rate in Job Runtime Estimates. In *2017 IEEE International Conference on Cluster Computing (CLUSTER)*. 530–540. https://doi.org/10.1109/CLUSTER.2017.11

[22] Dror G. Feitelson. 2015. *Workload Modeling for Computer Systems Performance Evaluation* (1st ed.). Cambridge University Press, USA.

[23] Ana Gainaru, Hongyang Sun, Guillaume Aupy, Yuankai Huo, Bennett A Landman, and Padma Raghavan. 2019. On-the-fly scheduling versus reservation-based scheduling for unpredictable workflows. *The International Journal of High Performance Computing Applications* 33, 6 (2019), 1140–1158. https://doi.org/10.1177/1094342019841681

[24] Mor Harchol-Balter. 2002. Task assignment with unknown duration. *J. ACM* 49, 2 (2002), 260–288.

[25] Mor Harchol-Balter, Mark E. Crovella, and Cristina D. Murta. 1999. On Choosing a Task Assignment Policy for a Distributed Server System. *J. Parallel and Distrib. Comput.* 59, 2 (1999), 204 – 228.

[26] Mor Harchol-Balter and Allen B. Downey. 1996. Exploiting Process Lifetime Distributions for Dynamic Load Balancing. In *Proceedings of the Int. Conf. on Measurement and Modeling of Computer Systems* (Philadelphia, Pennsylvania, USA) *(SIGMETRICS '96)*. ACM, New York, NY, USA, 13–24. https://doi.org/10.1145/233013.233019

[27] Mor Harchol-Balter, Alan Scheller-Wolf, and Andrew R. Young. 2009. Surprising Results on Task Assignment in Server Farms with High-variability Workloads. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems* (Seattle, WA, USA) *(SIGMETRICS '09)*. ACM, New York, NY, USA, 287–298.

[28] Retsef Levi, Thomas Magnanti, and Yaron Shaposhnik. 2019. Scheduling with Testing. *Management Science* 65, 2 (2019), 776–793. https://doi.org/10.1287/mnsc.2017.2973

[29] Alex F. Mills, Nilay Tanık Argon, and Serhan Ziya. 2013. Resource-Based Patient Prioritization in Mass-Casualty Incidents. *Manufacturing & Service Operations Management* 15, 3 (2013), 361–377. https://doi.org/10.1287/msom.1120.0426

[30] Mina Naghshnejad and Mukesh Singhal. 2020. A hybrid scheduling platform: a runtime prediction reliability aware scheduling platform to improve HPC scheduling performance. *The Journal of Supercomputing* 76 (2020), 28 pages.

[31] G. R. Nudd, D. J. Kerbyson, E. Papaefstathiou, S. C. Perry, J. S. Harper, and D. V. Wilcox. 2000. Pace—A Toolset for the Performance Prediction of Parallel and Distributed Systems. *The International Journal of High Performance Computing Applications* 14, 3 (2000), 228–251. https://doi.org/10.1177/109434200001400306

[32] Alan Scheller-Wolf and Karl Sigman. 1997. Delay Moments for FIFO GI/GI/s Queues. *Queueing Systems* 25, 1/4 (jan 1997), 77–95.

[33] Dan Tsafrir, Yoav Etsion, and Dror G. Feitelson. 2007. Backfilling Using System-Generated Predictions Rather than User Runtime Estimates. *IEEE Transactions on Parallel and Distributed Systems* 18, 6 (2007), 789–803. https://doi.org/10.1109/TPDS.2007.70606

[34] Y. Wiseman, K. Schwan, and P. Widener. 2004. Efficient end to end data exchange using configurable compression. In *24th Int. Conf. on Distributed Computing Systems*. 228–235. https://doi.org/10.1109/ICDCS.2004.1281587

[35] Carl Witt, Marc Bux, Wladislaw Gusew, and Ulf Leser. 2019. Predictive performance modeling for distributed batch processing using black box monitoring and machine learning. *Information Systems* 82 (2019), 33–52. https://doi.org/10.1016/j.is.2019.01.006

[36] Bo Zhang and Bert Zwart. 2013. Steady-State Analysis for Multiserver Queues Under Size Interval Task Assignment in the Quality-Driven Regime. *Math. of Op. Res.* 38, 3 (2013), 504–525. https://doi.org/10.1287/moor.1120.0571

[37] Darko Zivanovic, Milan Pavlovic, Milan Radulovic, Hyunsung Shin, Jongpil Son, Sally A. Mckee, Paul M. Carpenter, Petar Radojković, and Eduard Ayguadé. 2017. Main Memory in HPC: Do We Need More or Could We Live with Less? *ACM Trans. Archit. Code Optim.* 14, 1, Article 3 (mar 2017), 26 pages. https://doi.org/10.1145/3023362

[38] Salah Zrigui, Raphael Y. de Camargo, Arnaud Legrand, and Denis Trystram. 2022. Improving the performance of batch schedulers using online job runtime classification. *J. Parallel and Distrib. Comput.* 164 (2022), 83–95. https://doi.org/10.1016/j.jpdc.2022.01.003

A PROOFS

A.1 Proof of Proposition 1

First,

$$\min_{c\in[0,N]} R_c^{(N)}(\sigma) \le \min_{C\in\{1,\dots,N\}} R_C^{(N)}(\sigma) \tag{24}$$

and when C is an integer, (4) boils down to

$$R_C^{(N)}(\sigma) = \frac{\Lambda}{2}\frac{\mathbb{P}(Y_\sigma = x_m)^2\,\mathbb{E}[X^2 \mid Y_\sigma = x_m]}{C - \Lambda\,\mathbb{P}(Y_\sigma = x_m)\,\mathbb{E}[X \mid Y_\sigma = x_m]} + \frac{\Lambda}{2}\frac{\mathbb{P}(Y_\sigma = x_M)^2\,\mathbb{E}[X^2 \mid Y_\sigma = x_M]}{N - C - \Lambda\,\mathbb{P}(Y_\sigma = x_M)\,\mathbb{E}[X \mid Y_\sigma = x_M]} \tag{25}$$

provided that $\Lambda\,\mathbb{P}(Y_\sigma = x_m)\,\mathbb{E}[X \mid Y_\sigma = x_m] < C$ and $\Lambda\,\mathbb{P}(Y_\sigma = x_M)\,\mathbb{E}[X \mid Y_\sigma = x_M] < N - C$, and infinity otherwise. For any σ and N, the RHS term in (25) is strictly convex in C (provided that C is seen as a real number) and it is not difficult to see, by differentiation, that

$$\min_{C\in\{1,\dots,N\}} R_C^{(N)}(\sigma) = \min_{C\in\{\lfloor c^\star N\rfloor,\lfloor c^\star N\rfloor+1\}} R_C^{(N)}(\sigma) \tag{26}$$

where $c^\star(\sigma) := (1 - \lambda\mathbb{P}(Y_\sigma = x_M)\,\mathbb{E}[X \mid Y_\sigma = x_M])\,\gamma_m^* + \lambda\mathbb{P}(Y_\sigma = x_m)\,\mathbb{E}[X \mid Y_\sigma = x_m]\gamma_M^*$, with

$$\gamma_k^* := \gamma_k^*(\sigma) := \frac{\mathbb{P}(Y_\sigma = x_k)\sqrt{\mathbb{E}[X^2 \mid Y_\sigma = x_k]}}{\mathbb{E}[\sqrt{\mathbb{E}[X^2 \mid Y_\sigma]}]}, \quad k = m, M. \tag{27}$$

Substituting $\gamma_M^* = 1 - \gamma_m^*$ in $c^\star(\sigma)$, it can be written as

$$c^\star(\sigma) = \gamma_m^*(1 - \lambda\mathbb{E}[X]) + \lambda\mathbb{P}(Y_\sigma = x_m)\,\mathbb{E}[X \mid Y_\sigma = x_m] \tag{28}$$

and we notice that $c^\star(0) = c^*$ (see (8)). Now, let $(C_N)_N$ be a sequence in the form given in (7) and let also $\Delta_k := \mathbb{I}_{\{k=m\}} - \mathbb{I}_{\{k=M\}}$. We obtain

$$\begin{aligned}
&\min_{C\in\{\lfloor c^\star N\rfloor,\lfloor c^\star N\rfloor+1\}} R_C^{(N)}(\sigma) \le R_{C_N}^{(N)}(\sigma)\\
&= \frac{\lambda}{2}\frac{\mathbb{P}(Y_\sigma = x_m)^2\,\mathbb{E}[X^2 \mid Y_\sigma = x_m]}{c^\star + \frac{h_N}{N} - \lambda\,\mathbb{P}(Y_\sigma = x_m)\,\mathbb{E}[X \mid Y_\sigma = x_m]} + \frac{\lambda}{2}\frac{\mathbb{P}(Y_\sigma = x_M)^2\,\mathbb{E}[X^2 \mid Y_\sigma = x_M]}{1 - c^\star - \frac{h_N}{N} - \lambda\,\mathbb{P}(Y_\sigma = x_M)\,\mathbb{E}[X \mid Y_\sigma = x_M]}\\
&= \sum_{k=m,M}\frac{\lambda}{2\gamma_k^*}\frac{\mathbb{P}(Y_\sigma = x_k)^2\,\mathbb{E}[X^2 \mid Y_\sigma = x_k]}{1 + \frac{h_N}{\gamma_k^* N}\Delta_k - \lambda\mathbb{P}(Y_\sigma = x_M)\,\mathbb{E}[X \mid Y_\sigma = x_M]) - \lambda\,\mathbb{P}(Y_\sigma = x_m)\,\mathbb{E}[X \mid Y_\sigma = x_m]}\\
&= \sum_{k=m,M}\frac{\lambda}{2\gamma_k^*}\frac{\mathbb{P}(Y_\sigma = x_k)^2\,\mathbb{E}[X^2 \mid Y_\sigma = x_k]}{1 + \frac{h_N}{\gamma_k^* N}\Delta_k - \lambda\mathbb{E}[X]} && (29)\\
&= \sum_{k=m,M}\mathbb{P}(Y_\sigma = x_k)\sqrt{\mathbb{E}[X^2 \mid Y_\sigma = x_k]}\,\frac{\lambda}{2}\frac{\sum_{k'=m,M}\mathbb{P}(Y_\sigma = x'_k)\sqrt{\mathbb{E}[X^2 \mid Y_\sigma = x'_k]}}{1 + \frac{h_N}{\gamma_k^* N}\Delta_k - \lambda\mathbb{E}[X]}\\
&= \frac{\lambda}{2}\mathbb{E}[\sqrt{\mathbb{E}[X^2 \mid Y_\sigma]}]\sum_{k=m,M}\frac{\mathbb{P}(Y_\sigma = x_k)\sqrt{\mathbb{E}[X^2 \mid Y_\sigma = x_k]}}{1 + \frac{h_N}{\gamma_k^* N}\Delta_k - \lambda\mathbb{E}[X]} && (30)
\end{aligned}$$

$$\xrightarrow[N\to\infty]{} \frac{\lambda}{2}\frac{\mathbb{E}[\sqrt{\mathbb{E}[X^2 \mid Y_0]}]^2}{1-\lambda\mathbb{E}[X]}. \tag{31}$$

In (29) we have used the law of total expectation in (30) we have used the definition of conditional expectation, and in (31) we have used that both h_N/N and σ converge to zero as $N \to \infty$ (recall (9)).

To conclude the proof, it is enough to show that there exists a lower bound on $\min_{c\in[0,N]} R_c^{(N)}(\sigma)$ that matches (31) in the limit. This can be easily achieved by ignoring the $\frac{\Lambda}{2}\frac{p_z^2\mathbb{E}[Z^2]}{1-\Lambda p_z\mathbb{E}[Z]}$ term in (4) and using that $p_m \geq \frac{c-1}{c}$ and $p_M \geq \frac{N-c-2}{N-c}$.

A.2 Proof of Theorem 1

For simplicity, we assume that $C_N \in \mathbb{N}$ for all N. This slightly simplifies the proof because the term $\frac{p_z^2\mathbb{E}[Z^2]}{1-\Lambda p_z\mathbb{E}[Z]}$ in (4) disappears; note that this term is of the order of $1/N$ if C_N is not an integer. For $x \in \{x_m, x_M\}$, let $\Phi_x(\sigma) := \mathbb{P}(X = x \mid Y_\sigma = x)$. Let also $T := T(\sigma) := \mathbb{E}\left[\sqrt{\mathbb{E}[X^2 \mid Y_\sigma]}\right]$ and, for $k = m, M$, $T_k := T_k(\sigma) := \sqrt{\mathbb{E}[X^2 \mid Y_\sigma = x_k]}$. First,

$$\lim_{\sigma\downarrow 0}\frac{\mathrm{d}}{\mathrm{d}\sigma}D_{C_N}^{(N)}(\sigma) = f'(0) + \lim_{\sigma\downarrow 0}\frac{\mathrm{d}}{\mathrm{d}\sigma}R_{C_N}^{(N)}(\sigma). \tag{32}$$

Then, using (30) and since $\frac{h_N}{Y_k^* N}$ does not depend on σ,

$$\lim_{N\to\infty}\lim_{\sigma\downarrow 0}\frac{\mathrm{d}}{\mathrm{d}\sigma}R_{C_N}^{(N)}(\sigma) = \frac{\lambda}{2}\lim_{N\to\infty}\lim_{\sigma\downarrow 0}\frac{\mathrm{d}}{\mathrm{d}\sigma}\sum_{k=m,M}\frac{\mathbb{P}(Y_\sigma = x_k)T_k T}{1+\frac{h_N}{Y_k^* N}\Delta_k - \lambda\mathbb{E}[X]} \tag{33a}$$

$$= \frac{\lambda}{2}\lim_{\sigma\downarrow 0}\frac{\mathrm{d}}{\mathrm{d}\sigma}\sum_{k=m,M}\frac{\mathbb{P}(Y_\sigma = x_k)T_k T}{1-\lambda\mathbb{E}[X]} \tag{33b}$$

$$= \frac{\lambda}{2}\lim_{\sigma\downarrow 0}\frac{\mathrm{d}}{\mathrm{d}\sigma}\frac{T^2}{1-\lambda\mathbb{E}[X]} = \lambda\lim_{\sigma\downarrow 0}\frac{TT'}{1-\lambda\mathbb{E}[X]}. \tag{33c}$$

Since X and Y_0 are independent and identically distributed, $T_k(0) = T(0) = \sqrt{\mathbb{E}[X^2]}$ and we obtain

$$\begin{aligned}\lim_{\sigma\downarrow 0} T' &= \lim_{\sigma\downarrow 0}\sum_{k=m,M}\left(T_k\frac{\mathrm{d}\mathbb{P}(Y_\sigma = x_k)}{\mathrm{d}\sigma} + \mathbb{P}(Y_\sigma = x_k)T_k'\right)\\ &= \lim_{\sigma\downarrow 0}\sum_{k=m,M}\left(\sqrt{\mathbb{E}[X^2]}\frac{\mathrm{d}\mathbb{P}(Y_\sigma = x_k)}{\mathrm{d}\sigma} + \mathbb{P}(Y_\sigma = x_k)T_k'\right) = \sum_{k=m,M}\mathbb{P}(X = x_k)\lim_{\sigma\downarrow 0}T_k'\end{aligned} \tag{34}$$

with

$$\begin{aligned}2\sqrt{\mathbb{E}[X^2]}\lim_{\sigma\downarrow 0}T_m' &= \lim_{\sigma\downarrow 0}\frac{\mathrm{d}\mathbb{E}[X^2 \mid Y_\sigma = x_m]}{\mathrm{d}\sigma}\\ &= \sum_{k=m,M}x_k^2\lim_{\sigma\downarrow 0}\frac{\mathrm{d}}{\mathrm{d}\sigma}\mathbb{P}(X = x_k \mid Y_\sigma = x_m) = (x_m^2 - x_M^2)\Phi_{x_m}'(0)\end{aligned}$$

and, similarly

$$2\sqrt{\mathbb{E}[X^2]}\lim_{\sigma\downarrow 0}T_M' = (x_M^2 - x_m^2)\Phi_{x_M}'(0).$$

Thus, substituting back in (34)

$$T' \xrightarrow[\sigma\downarrow 0]{} (x_M^2 - x_m^2)\frac{\mathbb{P}(X = x_M)\Phi_{x_M}'(0) - \mathbb{P}(X = x_m)\Phi_{x_m}'(0)}{2\sqrt{\mathbb{E}[X^2]}}.$$

Since $\sum_{x,y} P'_{x,y} = 0$, and since X and Y_0 are independent and equal in distribution, for the conditional derivatives we obtain

$$\mathbb{P}(X = x_m)\Phi'_{x_m}(0) = P'_{x_m,x_m}(0) - \mathbb{P}(X = x_m)(P'_{x_m,x_m}(0) + P'_{x_M,x_m}(0)) \tag{35a}$$

$$\mathbb{P}(X = x_M)\Phi'_{x_M}(0) = P'_{x_M,x_M}(0) + \mathbb{P}(X = x_M)(P'_{x_m,x_m}(0) + P'_{x_M,x_m}(0)). \tag{35b}$$

Taking their difference, we obtain

$$\begin{aligned}\mathbb{P}(X = x_m)\Phi'_{x_m}(0) - \mathbb{P}(X = x_M)\Phi'_{x_M}(0) &= P'_{x_m,x_m}(0) - P'_{x_M,x_M}(0) - (P'_{x_m,x_m}(0) + P'_{x_M,x_m}(0)) \\ &= P'_{x_m,x_m}(0) + P'_{x_m,x_M}(0) = 0 \end{aligned}\tag{36}$$

where the last follows because $\mathbb{P}(X = x_m) = P_{x_m,x_m}(\sigma) + P_{x_m,x_M}(\sigma)$ is constant in σ. By (35a), we have thus shown that $T' \to 0$ as $\sigma \downarrow 0$ and that (see (33))

$$\lim_{N\to\infty} \lim_{\sigma\downarrow 0} \frac{\mathrm{d}}{\mathrm{d}\sigma} R^{(N)}_{C_N}(\sigma) = 0. \tag{37}$$

The proof is thus concluded by (32).

A.3 Proof of Theorem 2

As in the proof of Theorem 1, for simplicity we assume that $C_N \in \mathbb{N}$ for all N. Recall the definitions of T and T_k given in the proof of Theorem 1. Using (33)

$$\lim_{N\to\infty} \lim_{\sigma\downarrow 0} \frac{\mathrm{d}}{\mathrm{d}\sigma} D^{(N)}_{C_N}(\sigma) = f'(0) + \frac{\lambda\,\mathbb{E}[\sqrt{\mathbb{E}[X^2|Y_0]}]}{1 - \lambda\mathbb{E}[X]} \lim_{\sigma\downarrow 0} T'. \tag{38}$$

For T', using (13), we obtain

$$T' = \frac{\mathrm{d}}{\mathrm{d}\sigma}\left(x_m \mathbb{P}(Y_\sigma = x_m) + \mathbb{P}(Y_\sigma = x_M)\sqrt{x_m^2 + (x_M^2 - x_m^2)\frac{\mathbb{P}(X = x_M)}{\mathbb{P}(Y_\sigma = x_M)}}\right) \tag{39a}$$

$$= P'_{x_m,x_m}(\sigma)\left(x_m - \frac{\sqrt{\mathbb{E}[X^2] - x_m^2 P_{x_m,x_m}(\sigma)}}{2\sqrt{1 - P_{x_m,x_m}(\sigma)}} - \frac{x_m^2\sqrt{1 - P_{x_m,x_m}(\sigma)}}{2\sqrt{\mathbb{E}[X^2] - x_m^2 P_{x_m,x_m}(\sigma)}}\right) \tag{39b}$$

$$\xrightarrow[\sigma\downarrow 0]{} - P'_{x_m,x_m}(0)\left(\frac{\mathbb{E}[X^2 \mid Y_0 = x_M] + x_m^2}{2\sqrt{\mathbb{E}[X^2 \mid Y_0 = x_M]}} - x_m\right) \tag{39c}$$

and substituting in (38) gives (14). Now, for $\lim_{\sigma\downarrow 0} TT'$ we observe that

$$\begin{aligned}\lim_{\sigma\downarrow 0} T\,T' &= -P'_{x_m,x_m}(0)\left(x_m P_{x_m,x_m}(0) + \sqrt{1 - P_{x_m,x_m}(0)}\sqrt{\mathbb{E}[X^2] - x_m^2 P_{x_m,x_m}(0)}\right) \\ &\quad\times\left(\frac{\mathbb{E}[X^2] + x_m^2 - 2x_m^2 P_{x_m,x_m}(0)}{2\sqrt{1 - P_{x_m,x_m}(0)}\sqrt{\mathbb{E}[X^2] - x_m^2 P_{x_m,x_m}(0)}} - x_m\right) \\ &< -P'_{x_m,x_m}(0)\sqrt{\mathbb{E}[X^2]}\left(\frac{\mathbb{E}[X^2] + x_m^2}{2\sqrt{\mathbb{E}[X^2]}} - x_m\right)\end{aligned}$$

where the last inequality follows because $P'_{x_m,x_m} \geq 0$ (by Assumption 1) and because the mapping

$$z \mapsto \left(x_m z + \sqrt{1-z}\sqrt{\mathbb{E}[X^2] - x_m^2 z}\right)\left(\frac{\mathbb{E}[X^2] + x_m^2 - 2x_m^2 z}{2\sqrt{1-z}\sqrt{\mathbb{E}[X^2] - x_m^2 z}} - x_m\right)$$

is increasing in z over $[0,1]$. Substituting in (38) and rearranging terms, $\lim_{N\to\infty} \lim_{\sigma\downarrow 0} \frac{\mathrm{d}}{\mathrm{d}\sigma} D_{C_N}^{(N)}(\sigma) < 0$ if (15) holds true.

A.4 Proof of Proposition 2

Assume that $1 < c < N-1$. Let $v_m := \mathbb{P}(Y_\sigma = x_m) p_m \mathbb{E}[X \mid Y_\sigma = x_m]/\lfloor c \rfloor$, $v_z := p_z \mathbb{E}[Z]$ and $v_M := \mathbb{P}(Y_\sigma = x_M) p_M \mathbb{E}[X \mid Y_\sigma = x_M]/(N-1-\lfloor c \rfloor)$. Note that

$$\lfloor c \rfloor v_m + v_z + (N-1-\lfloor c \rfloor) v_M = \mathbb{E}[X \mid Y_\sigma = x_m]\mathbb{P}(Y_\sigma = x_m) + \mathbb{E}[X \mid Y_\sigma = x_M]\mathbb{P}(Y_\sigma = x_M) = \mathbb{E}[X] \quad (40)$$

where the last equality follows by the law of total expectation.

Now, using Jensen's inequality, we obtain

$$\frac{2}{\Lambda} R_c^{(N)}(\sigma) \geq \lfloor c \rfloor \frac{v_m^2}{1-\Lambda v_m} + \frac{v_z^2}{1-\Lambda v_z} + (N-1-\lfloor c \rfloor)\frac{v_M^2}{1-\Lambda v_M}.$$

Let $\mathcal{S}$ denote the set of $(v_m, v_z, v_M) \in [0,1]^3$ such that (40) holds and $v_m = v_z = v_M$. Since the mapping $s \mapsto \frac{u^2}{1-\Lambda u}$ is convex, by Karamata's inequality,

$$\frac{2}{\Lambda} R_c^{(N)}(\sigma) \geq \min_{(v_m,v_z,v_M)\in\mathcal{S}} \left\{\lfloor c \rfloor \frac{v_m^2}{1-\Lambda v_m} + \frac{v_z^2}{1-\Lambda v_z} + (N-1-\lfloor c \rfloor)\frac{v_M^2}{1-\Lambda v_M}\right\} = \frac{1}{N}\frac{\mathbb{E}[X]^2}{1-\lambda \mathbb{E}[X]}.$$

This concludes the proof.

A.5 Proof of Proposition 3

First, (1) implies

$$\sigma^* = \Theta(x_M^{1-\beta})$$

and, together with Assumption 1 and since $\beta < 1$,

$$\lim_{x_M\to\infty} P(\sigma^*) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \quad (41)$$

For $p = 1, 2$, we notice

$$\mathbb{P}(Y_{\sigma^*} = x_m)\mathbb{E}[X^p \mid Y_{\sigma^*} = x_m]^{\frac{1}{p}} = \mathbb{P}(Y_{\sigma^*} = x_m)^{1-\frac{1}{p}} \left(x_m^p P_{x_m,x_m}(\sigma^*) + x_M^p P_{x_M,x_m}(\sigma^*)\right)^{\frac{1}{p}} \xrightarrow[x_M\to\infty]{} x_m \quad (42)$$

because $x_M^2 P_{x_M,x_m}(\sigma^*) = x_M^2 P_{x_M,x_m}(\Theta(x_M^{1-\beta})) = x_M^2\, O(x_M^{-(1-\beta)\eta_m}) \to 0$ as $x_M \to \infty$. Now, since $1 = \mathbb{P}(X = x_m) + P_{x_M,x_m} + P_{x_M,x_M} = 1 - \alpha x_M^{-\beta} + P_{x_M,x_m} + P_{x_M,x_M}$, we obtain

$$P_{x_M,x_M}(\sigma^*) = \alpha x_M^{-\beta} - O\left(x_M^{-\eta_m(1-\beta)}\right),$$

and since $\eta_m > \frac{2}{1-\beta}$,

$$\lim_{x_M\to\infty} \frac{P_{x_M,x_M}(\sigma^*)}{\alpha x_M^{-\beta}} = 1. \tag{43}$$

Thus,

$$\mathbb{P}(Y_{\sigma^*} = x_M)\frac{\mathbb{E}[X \mid Y_{\sigma^*} = x_M]}{\mathbb{E}[X]} = \frac{1}{\mathbb{E}[X]}\left(x_m P_{x_m,x_M}(\sigma^*) + x_M P_{x_M,x_M}(\sigma^*)\right) \xrightarrow[x_M\to\infty]{} 1 \tag{44}$$

and similarly

$$\lim_{x_M\to\infty} \mathbb{P}(Y_{\sigma^*} = x_M)\frac{\sqrt{\mathbb{E}[X^2 \mid Y_{\sigma^*} = x_M]}}{\mathbb{E}[X]} \tag{45a}$$

$$= \lim_{x_M\to\infty} \frac{1}{\mathbb{E}[X]}\sqrt{\mathbb{P}(Y_{\sigma^*} = x_M)}\sqrt{x_m^2 P_{x_m,x_M}(\sigma^*) + x_M^2 P_{x_M,x_M}(\sigma^*)} \tag{45b}$$

$$= \lim_{x_M\to\infty} \frac{x_M}{\mathbb{E}[X]}\sqrt{P_{x_M,x_M}(\sigma^*)}\sqrt{P_{x_m,x_M}(\sigma^*)} + \frac{1}{\mathbb{E}[X]} \times x_M P_{x_M,x_M}(\sigma^*) = 1. \tag{45c}$$

In addition, from (45), $\lim_{x_M\to\infty} \mathbb{P}(Y_{\sigma^*} = x_M)\sqrt{\mathbb{E}[X^2 \mid Y_{\sigma^*} = x_M]} = +\infty$, which implies

$$\lim_{x_M\to\infty} c^\star = \lim_{x_M\to\infty} (1-\rho)\frac{\sqrt{\mathbb{E}[X^2 \mid Y_{\sigma^\star} = x_m]}}{\mathbb{E}[\sqrt{\mathbb{E}[X^2 \mid Y_{\sigma^\star}]}]}\mathbb{P}(Y_{\sigma^\star} = x_m) + \rho\, x^{\beta-1}\mathbb{E}[X \mid Y_{\sigma^\star} = x_m]\mathbb{P}(Y_{\sigma^\star} = x_m)$$

$$= \frac{(1-\rho)x_m}{x_m + \lim_{x_M\to\infty} \mathbb{P}(Y_{\sigma^*} = x_M)\sqrt{\mathbb{E}[X^2 \mid Y_{\sigma^*} = x_M]}} = 0.$$

If $N(1-\rho) > 1$, combining the properties above we obtain

$$\lim_{x_M\to\infty} \frac{D_c^{(N)}(\sigma^*)}{R\downarrow} = \lim_{x_M\to\infty} \frac{1}{R\downarrow}\left(\frac{\sigma^*}{1-\Lambda\sigma^*} + R_c^{(N)}(\sigma^*)\right) = \frac{1}{\gamma} + \lim_{x_M\to\infty} \frac{R_1^{(N)}(\sigma^*)}{R\downarrow}$$

$$= \frac{1}{\gamma} + \lim_{x_M\to\infty} \frac{\Lambda}{2R\downarrow}\left(\frac{\mathbb{P}(Y_{\sigma^*} = x_m)^2\,\mathbb{E}[X^2 \mid Y_\sigma^* = x_m]}{1-\Lambda\mathbb{P}(Y_{\sigma^*} = x_m)\,\mathbb{E}[X \mid Y_\sigma^* = x_m]} + \frac{\mathbb{P}(Y_{\sigma^*} = x_M)^2\mathbb{E}[X^2 \mid Y_\sigma^* = x_M]}{N-1-\Lambda\mathbb{P}(Y_{\sigma^*} = x_M)\,\mathbb{E}[X \mid Y_\sigma^* = x_M]}\right)$$

$$= \frac{1}{\gamma} + \lim_{x_M\to\infty} \frac{\rho N}{2R\downarrow\mathbb{E}[X]}\left(\frac{x_m^2}{1-\frac{\rho}{\mathbb{E}[X]}Nx_m} + \frac{\mathbb{P}(Y_{\sigma^*} = x_M)^2\mathbb{E}[X^2 \mid Y_\sigma^* = x_M]}{N-1-\rho N\frac{\mathbb{P}(Y_{\sigma^*}=x_M)\,\mathbb{E}[X \mid Y_\sigma^*=x_M]}{\mathbb{E}[X]}}\right)$$

$$= \frac{1}{\gamma} + \lim_{x_M\to\infty} \frac{\rho N}{2R\downarrow\mathbb{E}[X]} \times \frac{\mathbb{P}(Y_{\sigma^*} = x_M)^2\mathbb{E}[X^2 \mid Y_\sigma^* = x_M]}{N(1-\rho)-1}$$

$$= \frac{1}{\gamma} + \frac{N(1-\rho)}{N(1-\rho)-1}$$

This proves (19). Therefore, assume that $N(1-\rho) \le 1$. In this case, $\lim_{x_M\to\infty} c = N(1-\rho) \le 1$ and therefore no server will receive jobs that are only predicted as short. Now, we notice that

$$\frac{p_z\mathbb{E}[Z]}{\mathbb{E}[X]} = 1 - \frac{N-1}{N-c}\mathbb{P}(Y_{\sigma^*} = x_M)\,\frac{\mathbb{E}[X \mid Y_{\sigma^*} = x_M]}{\mathbb{E}[X]}$$

$$= 1 - \frac{N-1}{N\rho + N\rho\,\phi(x_M)}\mathbb{P}(Y_{\sigma^*} = x_M)\,\frac{\mathbb{E}[X \mid Y_{\sigma^*} = x_M]}{\mathbb{E}[X]} \xrightarrow[x_M\to\infty]{} 1 - \frac{N-1}{N\rho}$$

where in the last expression we have used (44). Similarly,

$$\frac{p_z^2\mathbb{E}[Z^2]}{\mathbb{E}[X^2]} = p_z\left(1 - \frac{N-1}{N-c}\mathbb{P}(Y_{\sigma^*} = x_M)\frac{\mathbb{E}[X^2 \mid Y_{\sigma^*} = x_M]}{\mathbb{E}[X^2]}\right) \xrightarrow[x_M\to\infty]{} 1 - \frac{N-1}{N\rho}$$

Using these limits,

$$\begin{aligned}
\lim_{x_M\to\infty} \frac{D_c^{(N)}(\sigma^*)}{x_M^\theta R^{\downarrow}} &= \lim_{x_M\to\infty} \frac{R_c^{(N)}(\sigma^*)}{x_M^\theta R^{\downarrow}} \\
&= \lim_{x_M\to\infty} \frac{\Lambda}{2x_M^\theta R^{\downarrow}} \left(\frac{p_z^2 \mathbb{E}[Z^2]}{1-\Lambda p_z\,\mathbb{E}[Z]} + \frac{N-1}{N-c} \frac{\mathbb{P}(Y_{\sigma^*}=x_M)^2 \mathbb{E}[X^2 \mid Y_{\sigma^*}=x_M]}{N-c-\Lambda \mathbb{P}(Y_{\sigma^*}=x_M)\,\mathbb{E}[X \mid Y_{\sigma^*}=x_M]} \right) \\
&= \lim_{x_M\to\infty} \frac{\rho N}{2x_M^\theta \mathbb{E}[X] R^{\downarrow}} \left(\frac{p_z^2 \mathbb{E}[Z^2]}{1-\rho N \frac{p_z\,\mathbb{E}[Z]}{\mathbb{E}[X]}} + \frac{N-1}{N-c}\frac{\mathbb{E}[X]^2}{N-c-\rho N} \right) \\
&= \lim_{x_M\to\infty} \frac{\rho N}{2x_M^\theta \mathbb{E}[X] R^{\downarrow}} \left(\frac{p_z^2 \mathbb{E}[Z^2]}{N(1-\rho)} + \frac{N-1}{N\rho + N\rho\,\phi(x_M)} \frac{\mathbb{E}[X]^2}{N\rho\,\phi(x_M)} \right) \\
&= \lim_{x_M\to\infty} \frac{1-\rho}{x_M^\theta \mathbb{E}[X]^2} \left(\frac{p_z^2 \mathbb{E}[Z^2]}{1-\rho} + \frac{N-1}{\rho + \rho\,\phi(x_M)} \frac{\mathbb{E}[X]^2}{N\rho\,\phi(x_M)} \right) \\
&= \left(1 - \frac{N-1}{N\rho}\right) \lim_{x_M\to\infty} x_M^{-\theta} + (1-\rho) \lim_{x_M\to\infty} \frac{N-1}{\rho+\rho\,\phi(x_M)} \frac{1}{N\rho\,\phi(x_M)\,x_M^\theta} = (1-\rho)\frac{N-1}{N\rho^2}
\end{aligned}$$

and this proves (20).

A.6 Proof of Theorem 3

Given Proposition 3, it is enough to show that $R^{\downarrow}/\min_{c\in[0,N]} R_c^{(N)}(0) \to 0$ as $x_M \to \infty$. Let $\mathcal{S}$ be the set of $c \in [1, N-1]$ such that $R_c^{(N)}(0) < \infty$.

For now, let us assume that X and Y_0 are independent. In this case, the mean user-perceived waiting time corresponds to the mean response time of a single M/G/1 queue. Using Pollaczek–Khinchine formula, this is given by

$$\min_{c\in[0,N]} R_c^{(N)}(0) = \frac{\rho}{2\mathbb{E}[X]} \frac{\mathbb{E}[X^2]}{1-\lambda\mathbb{E}[X]} \tag{46}$$

Since both $\frac{\mathbb{E}[X^2]}{x_M \mathbb{E}[X]}$ and $R^{\downarrow} x_M^{1-\beta}$ converge to a strictly positive constant as $x_M \to \infty$,

$$\frac{R^{\downarrow}}{\min_{c\in[0,N]} R_c^{(N)}(0)} = O(x_M^{-\beta})$$

and this gives (21) in case i).

Now, assume in the remainder that (12) holds. Let $\underline{c} := \Lambda \mathbb{P}(Y_0 = x_m)\mathbb{E}[X \mid Y_0 = x_m]$ and notice that $\underline{c}$ makes equal to one the load of servers that only receives jobs that are predicted as short. Let also

$$R_{c,M}^{(N)} := \frac{\Lambda}{2}\left(\frac{\mathbb{P}(Y_0=x_M)^2 (1-p_M)^2 \mathbb{E}[X^2 \mid Y_0=x_M]}{1-\Lambda\mathbb{P}(Y_0=x_M)\,(1-p_M)\,\mathbb{E}[X \mid Y_0=x_M]} + \frac{\mathbb{P}(Y_0=x_M)^2 p_M^2 \mathbb{E}[X^2 \mid Y_0=x_M]}{N-1-\lfloor c \rfloor - \Lambda \mathbb{P}(Y_0=x_M) p_M\,\mathbb{E}[X \mid Y_0=x_M]} \right) \tag{47}$$

which in view of (4) can be interpreted as the mean waiting time at servers that is due to long-predicted jobs only (i.e., as if jobs that are predicted as short are discarded). Then,

$$\min_{c\in[0,N]} R_c^{(N)}(0) \ge \min_{c\in\mathcal{S}} R_{c,M}^{(N)} = \lim_{c\uparrow\underline{c}} R_{c,M}^{(N)} \tag{48a}$$

$$\ge \frac{\Lambda}{2} \frac{\mathbb{P}(Y_0=x_M)^2 p_M^2 \mathbb{E}[X^2 \mid Y_0=x_M]}{N-1-\lfloor \underline{c} \rfloor - \Lambda\mathbb{P}(Y_0=x_M) p_M\,\mathbb{E}[X \mid Y_0=x_M]} \tag{48b}$$

$$= \frac{\Lambda}{2} \frac{N-1-\lfloor \underline{c} \rfloor}{N-\underline{c}} \frac{\mathbb{P}(Y_0 = x_M)^2 \mathbb{E}[X^2 \mid Y_0 = x_M]}{N - \Lambda \mathbb{E}[\mathbb{E}[X \mid Y_0]]} \tag{48c}$$

$$= \frac{\rho}{2\mathbb{E}[X]} \frac{N-1-\lfloor \underline{c} \rfloor}{N-\underline{c}} \frac{\mathbb{P}(Y_0 = x_M)^2 \mathbb{E}[X^2 \mid Y_0 = x_M]}{1 - \lambda \mathbb{E}[X]} \tag{48d}$$

By (13), $\mathbb{P}(Y_0 = x_m) = P_{x_m,x_m}(0)$, $\mathbb{P}(Y_0 = x_M) \geq \mathbb{P}(X = x_M)$,

$$\begin{aligned}
&\mathbb{P}(Y_0 = x_M)^2 \mathbb{E}[X^2 \mid Y_0 = x_M] = \mathbb{P}(Y_0 = x_M)^2 (x_m^2 \mathbb{P}(X = x_m \mid Y_0 = x_M) + x_M^2 \mathbb{P}(X = x_M \mid Y_0 = x_M)) \\
&= \mathbb{P}(Y_0 = x_M)(x_m^2 \mathbb{P}(X = x_m \cap Y_0 = x_M) + x_M^2 \mathbb{P}(X = x_M \cap Y_0 = x_M)) \\
&= \mathbb{P}(Y_0 = x_M)(x_m^2 \mathbb{P}(X = x_m \cap Y_0 = x_M) + x_M^2 \mathbb{P}(X = x_M) \\
&= \mathbb{P}(Y_0 = x_M)(x_m^2 (\mathbb{P}(X = x_m \cap Y_0 = x_M) + \mathbb{P}(X = x_M \cap Y_0 = x_M)) + (x_M^2 - x_m^2)\mathbb{P}(X = x_M) \\
&= \mathbb{P}(Y_0 = x_M) \left(x_m^2 \mathbb{P}(Y_0 = x_M) + (x_M^2 - x_m^2)\mathbb{P}(X = x_M) \right) \\
&= (1 - P_{x_m,x_m}(0)) \left(\mathbb{E}[X^2] - x_m^2 P_{x_m,x_m}(0) \right)
\end{aligned}$$

and, as $x_M \to \infty$, $\underline{c} = \rho N \mathbb{P}(Y_0 = x_m) \frac{\mathbb{E}[X \mid Y_0 = x_m]}{\mathbb{E}[X]} = \rho N \mathbb{P}(Y_0 = x_m) \frac{x_m}{\mathbb{E}[X]} \to 0$. Combining these properties in (48d), we have thus shown that

$$\min_{c \in [0,N]} R_c^{(N)}(0) \geq \frac{\rho}{2\mathbb{E}[X]} \frac{N-1-\lfloor \underline{c} \rfloor}{N-\underline{c}} \frac{(1 - P_{x_m,x_m}(0)) \left(\mathbb{E}[X^2] - x_m^2 P_{x_m,x_m}(0) \right)}{1-\rho} = (1 - P_{x_m,x_m}(0))\, \Theta(x_M).$$

Therefore, given the scaling assumptions on $P_{x_m,x_m}(0)$ we obtain

$$\frac{R^{\downarrow}}{\min\limits_{c \in [0,N]} R_c^{(N)}(0)} \leq \frac{1}{(1 - P_{x_m,x_m}(0))\, O(x_M^{\beta})} \xrightarrow[x_M \to \infty]{} 0,$$

which gives (21) in case ii).

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009