Performance analysis of work stealing in large scale multithreaded computing

NIKKI SONENBERG, The Alan Turing Institute, United Kingdom
GRZEGORZ KIELANSKI, University of Antwerp, Belgium
BENNY VAN HOUDT, University of Antwerp, Belgium

Randomized work stealing is used in distributed systems to increase performance and improve resource utilization. In this paper, we consider randomized work stealing in a large system of homogeneous processors where parent jobs spawn child jobs that can feasibly be executed in parallel with the parent job. We analyse the performance of two work stealing strategies: one where only child jobs can be transferred across servers; the other where parent jobs are transferred. We define a mean field model to derive the response time distribution in a large scale system with Poisson arrivals and exponential parent and child job durations. We prove that the model has a unique fixed point that corresponds to the steady state of a structured Markov chain, allowing us to use matrix analytic methods to compute the unique fixed point. The accuracy of the mean field model is validated using simulation. Using numerical examples we illustrate the effect of different probe rates, load, and different child job size distributions on performance with respect to the two stealing strategies, individually, and compared to each other.

CCS Concepts: • **Networks** → **Network performance modeling**.

Additional Key Words and Phrases: mean field model, matrix analytic methods, performance analysis, distributed computing

ACM Reference Format:
Nikki Sonenberg, Grzegorz Kielanski, and Benny Van Houdt. 2020. Performance analysis of work stealing in large scale multithreaded computing. 1, 1 (June 2020), 28 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Modern computer systems involve large amounts of parallelism, and the ways in which jobs and servers can interact have grown in complexity [11, 21]. One form of parallelism is multithreaded computation which involves a set of threads, each of which is a sequential ordering of tasks. Computation starts by executing a main thread (parent job), and a thread can create or spawn other threads (child jobs) that are initially stored locally but then can be migrated and executed on other servers [4, 26].

A longstanding approach to redistribute work among a set of processors is the concept of randomized work stealing [27], one that has been implemented in various systems such as Cilk [3], Intel TBB [20] and KAAPI [9]. The main idea is that processors that become idle attempt to *steal* work from another processor selected uniformly at random [4, 6]. An alternate approach where processors with pending tasks attempt to locate idle processors is known as load sharing.

Authors' addresses: Nikki Sonenberg, The Alan Turing Institute, London, United Kingdom, nsonenberg@turing.ac.uk; Grzegorz Kielanski, University of Antwerp, Antwerp, Belgium, grzegorz.kielanski@uantwerpen.be; Benny Van Houdt, University of Antwerp, Antwerp, Belgium, benny.vanhoudt@uantwerpen.be.

XXXX-XXXX/2020/6-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

In this paper, we consider a system of homogeneous processors operating with randomized work stealing and study the problem of balancing processor workloads. We consider two work stealing protocols: one where only child jobs are able to be migrated across servers; and one where parent jobs can be migrated across servers. We define mean field models for both stealing strategies and validate the models using simulation. We prove the existence of a unique fixed point for each model and use this to study the performance of these strategies. While we do not present convergence proofs for the stationary measures, such proofs for convergence over finite time scales can be constructed using existing mean field theory [7, 13], see Section 9 for more comments.

In this paper we make the following contributions:

- We present two mean field models for work stealing in multithreaded computations.
- We prove these models have a unique fixed point that can be computed efficiently using matrix analytic methods (by solving a single Quasi-Birth-Death Markov chain). This is the main technical contribution of the paper.
- We indicate how to compute the response time distribution of a job for both strategies.
- For both strategies we illustrate the effect on mean response time of varying probe rate, load and child job size distributions. In selected scenarios, we show that with high probe rate and low loads, child stealing achieves a lower mean response time; but the parent stealing strategy performs better under low probe rate and high loads. Somewhat surprisingly, we show that under high probe rates, the variability of the child job size distribution may improve performance.
- The developed model and methods provide a foundation for the performance analysis of more general systems with similar features.

The rest of this paper is organised as follows. Related work is discussed in Section 2, while in Section 3 we describe the system and the work stealing strategies considered. The mean field model is introduced in Section 4. The model of a single server is introduced in Section 5 and its associated response time distribution is derived in Section 6. The stationary distribution of the single server queue is shown to be the unique fixed point of the mean field model in Section 7. In Section 8 we present explicit results when the probe rate tends to infinity. We validate the mean field model using simulation in Section 9. The performance of the stealing strategies with numerical examples is presented in Section 10. Finally, Section 11 contains some concluding remarks and discusses some model generalizations.

2 RELATED WORK

Some initial analytical models developed to study the performance of work stealing and sharing strategies can be found in [6, 17, 23]. In [6] the authors consider a homogeneous system with exponential job sizes and compare the performance of work stealing and work sharing strategies. To generate numerical results the authors rely on a decoupling assumption combined with an iterative approach. As indicated in [15], this approach is equivalent to computing a fixed point of the drift equations of a mean field model. This work was further extended to heterogeneous systems in [17] using a similar approach. These studies demonstrate that work stealing is far more effective than work sharing when the system load is high, which explains why most practical systems rely on load stealing techniques. [23] focused on work stealing only and motivated by shared-memory systems assumes that migrated jobs have a higher service demand and migrating jobs requires some time. The analysis is based on a decomposition assumption and can therefore also be reformulated as a mean field approximation.

More recent work based on mean field models for work stealing and sharing includes [8, 15, 22, 25]. A model for stealing in a network composed of a number of homogeneous clusters and exponential

job sizes is presented in [8], where an important difference with prior work lies in the fact that half of the jobs are stolen instead of just one job at a time. The main motivation for the work presented in [15, 22, 25] was to provide a more fair comparison between load stealing and sharing strategies. For the more traditional strategies considered in [6, 17] the communication overhead of the stealing and sharing strategies is not the same, which made the comparison somewhat biased. Exponential job sizes were considered in [15, 22] for homogeneous and heterogeneous networks and homogeneous networks with non-exponential job sizes were considered in [25].

A common feature in all these prior works is that a job is always migrated between servers as a whole. The key novel feature in this paper is to allow part of a job to be transferred. More specifically, motivated by multi-threaded computing, our main interest lies in the analysis of a system in which parent jobs spawn child jobs when their service starts. These child jobs are initially stored locally and can subsequently be stolen by idle servers. An important contribution lies in the fact that our results indicate that we can approximate the performance of such a large-scale system closely using the steady state vector of a single structured Markov chain and thus without the need for an iterative procedure. Most of the paper is devoted to the analysis of the system with child job stealing.

We also compare the performance of this system with a system in which only parent jobs can be stolen. Note that in such case jobs are again transferred as a whole and therefore we can make use of the results in [25] by considering the parent job together with its spawned child jobs as a single non-exponential job. The analysis relies on the same overall approach for the parent and child job stealing systems: we define a mean field model and a structured Markov chain and show that the unique fixed point of the mean field model is the steady state of the structured Markov chain. However defining the structured Markov chain and proving the above result is much harder for the child job stealing system. In addition, the results obtained for the child job stealing system are more elegant as we managed to obtain an explicit expression for the rate at which idle servers steal jobs, which does not appear to be feasible for the system with parent job stealing (which involves solving a non-linear matrix equation numerically). Finally we also provide a means to compute the response time distribution, while prior work focused on the mean response time.

3 SYSTEM DESCRIPTION AND STRATEGIES

We consider a system with the following characteristics:

i. N homogeneous servers each with an infinite buffer to store jobs.
ii. Each server is subject to its own local Poisson arrival process with rate λ. Arriving jobs are referred to as parent jobs.
iii. Upon a parent entering service, the parent job spawns $i \in \{0, 1, \ldots, m\}, m \geq 1$, child jobs at that server, the number of which follows a general distribution with finite support, $\breve{p} = \{p_i\}$. We refer to a *job* as a parent job and its spawned child jobs.
iv. Child jobs spawned at a server are served before any *waiting* parent jobs are served, but can only start service at the local server after their parent job completes service.
v. It is assumed that parent and child jobs have exponentially distributed service requirements with rates μ_1 and μ_2, respectively.

In this paper we study the performance of rate based work stealing strategies [16, 25] in our model of multithreaded computations. More specifically we consider the following two randomized work stealing protocols:

- *Parent job stealing*. When a server is idle, it generates probe messages at rate r. As long as the server remains idle, probes are sent according to Poisson process with rate r. This process is interrupted whenever the server becomes busy. The probed server is selected at random and

a probe is successful if there are parent jobs *waiting* to be served. We assume the policy is to always steal the oldest parent job, that is, the head-of-the-line parent job in the waiting room. Note that such a parent job did yet not spawn child jobs.

- *Child job stealing*. Idle servers again probe at rate r, but a probe is only successful if there is at least one child job waiting to be served. In this case a single child job is transferred to the idle probing server. Extending the model such that multiple child jobs can be stolen at once is non-trivial and subject to future work.

We compare the performance of these two strategies, noting that when a parent job is transferred to an idle server, its service immediately starts and its child jobs are spawned at this server. Probes and job transfers are assumed to be instantaneous. Another interpretation of the rate based probing is that it takes an exponentially distributed amount of time with mean $1/r$ to probe another server (and to transfer the job if the probe is successful) and steal attempts are executed sequentially.

We discuss a number of relaxations such as phase-type service times, batch steal events and heterogeneous servers in Section 11.

4 MEAN FIELD MODEL

We use a mean field model to describe the system with $N \to \infty$ servers. The infinite system is defined by a set of ODEs and we use the superscript (c) when referring to the system where child job stealing is allowed and superscript (p) where parent job stealing is allowed.

For $i \in \{(c),(p)\}$, denote by $f^{i}_{\ell,j,k}(t)$ the fraction of servers with ℓ parent jobs *waiting* in the queue, $j \in \{0,1,\ldots,m\}$ child jobs in the queue and $k \in \{0,1\}$ describing whether a parent is in service ($k=1$) or not ($k=0$) at time t. Note that ℓ does not count parent jobs in service, whereas j counts child jobs waiting and in service. Let $f^{i}_{*}(t)$ be the fraction of idle queues at time t, that is when $\ell, j, k = 0$. Let $1[A]$ be equal to one if A is true and zero otherwise.

4.1 Child job stealing

We start by presenting the drift equations for the system where child jobs are stolen followed by a detailed discussion. For $\ell \geq 0$ and $j+k \geq 1$,

$$\begin{aligned}
\frac{d}{dt}f^{(c)}_{\ell,j,k}(t) &= \lambda f^{(c)}_{\ell-1,j,k}(t)1[\ell \geq 1] + \lambda p_j f^{(c)}_{*}(t)1[\ell = 0, k = 1] - \lambda f^{(c)}_{\ell,j,k}(t) + \mu_1 f^{(c)}_{\ell,j,k+1}(t)1[k=0] \\
&\quad + \mu_1 p_j f^{(c)}_{\ell+1,0,k}(t)1[k=1] - \mu_1 f^{(c)}_{\ell,j,k}(t)1[k=1] + \mu_2 f^{(c)}_{\ell,j+1,k}(t)1[j \leq m-1, k=0] \\
&\quad + \mu_2 p_j f^{(c)}_{\ell+1,1,k-1}(t)1[k=1] - \mu_2 f^{(c)}_{\ell,j,k}(t)1[k=0] + r f^{(c)}_{*}(t) f^{(c)}_{\ell,j+1,k}(t)1[j \leq m-1] \\
&\quad - r f^{(c)}_{*}(t) f^{(c)}_{\ell,j,k}(t)1[j+k>1] + r f^{(c)}_{*}(t) \sum_{\substack{\ell' \geq 0, \\ j'+k'>1}} f^{(c)}_{\ell',j',k'}(t)1[\ell = 0, j = 1, k = 0],
\end{aligned}$$

and for $\ell, j, k = 0$,

$$\frac{d}{dt}f^{(c)}_{*}(t) = -\lambda f^{(c)}_{*}(t) + \mu_1 f^{(c)}_{0,0,1}(t) + \mu_2 f^{(c)}_{0,1,0}(t) - r f^{(c)}_{*}(t) \sum_{\substack{\ell \geq 0, \\ j+k>1}} f^{(c)}_{\ell,j,k}(t).$$

The first three terms of the drift of $f^{(c)}_{\ell,j,k}(t)$ correspond to arrivals of parent jobs, in which we distinguish between arrivals to a non-idle server and to an idle server. The following three terms correspond to service completions of a parent job, distinguishing when the head of the queue is either a child or a parent job. The following three terms correspond to service completions of a child job, distinguishing when the head of the queue is either a child or a parent job. The remaining three terms correspond to child job transfers, with the last term capturing the successful transfer

to an idle server. The condition $1[j+k>1]$ guarantees that we have at least one child job waiting (as a child job in service is never stolen). Note that

$$\sum_{\ell\geq 0, j+k>1} f^{(c)}_{\ell,j,k}(t) = 1 - f^{(c)}_{*}(t) - \sum_{\ell\geq 0}(f^{(c)}_{\ell,1,0}(t) + f^{(c)}_{\ell,0,1}(t)),$$

which equals the probability that a probe transmitted at time t succeeds in stealing a child job. For the drift of $f^{(c)}_{*}(t)$, the first term is due to arrivals of parent jobs, the second and third due to parent and child job completions, respectively and the last term is due to child job transfers.

We now rewrite these equations in matrix form, using the vectors below, where 0_i is a column vector of zeroes of length i, e_i is the i-th row of the unit matrix and e a column vector of ones:

$$\begin{aligned} f^{(c)}_{\ell}(t) &= \left(f^{(c)}_{\ell,1,0}(t), \ldots, f^{(c)}_{\ell,m,0}(t), f^{(c)}_{\ell,0,1}(t), \ldots, f^{(c)}_{\ell,m,1}(t)\right), \\ \alpha &= \begin{bmatrix} 0'_m & p_0 & p_1 & \ldots & p_m \end{bmatrix}, \\ \mu &= \begin{bmatrix} \mu_2 & 0'_{m-1} & \mu_1 & 0'_m \end{bmatrix}', \\ v_0 &= \begin{bmatrix} 1 & 0'_{m-1} & 1 & 0'_m \end{bmatrix}', \end{aligned} \tag{1}$$

where $f^{(c)}_{\ell}(t)$ and α are row vectors of size $2m+1$, while μ and v_0 are column vectors of size $2m+1$. Note that v_0 marks the states where $j+k=1$, which are the states where there are no child jobs waiting for service. We then have for $\ell \geq 0$,

$$\begin{aligned} \frac{d}{dt} f^{(c)}_{\ell}(t) = {} & \lambda f^{(c)}_{\ell-1}(t) 1[\ell \geq 1] - \lambda f^{(c)}_{\ell}(t) + \lambda f^{(c)}_{*}(t)\alpha 1[\ell = 0] + f^{(c)}_{\ell}(t) S^{(c)}(r,t) + f^{(c)}_{\ell+1}(t)\mu\alpha \\ & + r f^{(c)}_{*}(t) \sum_{\ell'\geq 0} f^{(c)}_{\ell'}(t)(e - v_0) e_1 1[\ell = 0], \end{aligned} \tag{2}$$

and

$$\frac{d}{dt} f^{(c)}_{*}(t) = -\lambda f^{(c)}_{*}(t) + f^{(c)}_{0}(t)\mu - r f^{(c)}_{*}(t) \sum_{\ell\geq 0} f^{(c)}_{\ell}(t)(e - v_0). \tag{3}$$

The $(2m+1)\times(2m+1)$ matrix $S^{(c)}(r,t)$ is defined as

$$S^{(c)}(r,t) = \begin{bmatrix} S^{(c)}_{00}(r,t) & 0 \\ S_{10} & S^{(c)}_{11}(r,t) \end{bmatrix}, \tag{4}$$

with

$$S^{(c)}_{00}(r,t) = \begin{bmatrix} -\mu_2 & & & \\ \mu_2 + r f_*(t) & -(\mu_2 + r f_*(t)) & & \\ & \mu_2 + r f_*(t) & -(\mu_2 + r f_*(t)) & \\ & & \ddots & \end{bmatrix}, \quad S_{10} = \begin{bmatrix} 0 & \ldots & & \\ \mu_1 & & & \\ & \mu_1 & & \\ & & \mu_1 & \\ & & & \ddots \end{bmatrix},$$

$$S^{(c)}_{11}(r,t) = \begin{bmatrix} -\mu_1 & & & \\ r f_*(t) & -(\mu_1 + r f_*(t)) & & \\ & r f_*(t) & -(\mu_1 + r f_*(t)) & \\ & & \ddots & \end{bmatrix}.$$

The (off diagonal) entries of the matrix $S^{(c)}(r,t)$ corresponds to events that do not lead to a change in the value of $\ell \geq 1$. The matrix is partitioned according to the type of job in service following an event: $S^{(c)}_{00}(r,t)$ captures state changes where a child job remains in service (either a child is

stolen from the queue, or the child job in service completes and a waiting child job enters service); S_{10} captures the event where a parent job completes and a child job starts service; and $S_{11}^{(c)}(r,t)$ captures state changes where the parent job remains in service and a child job is stolen.

4.2 Parent job stealing

This mean field model is a special case of the one described in [25] by considering the parent and its child jobs as a single non-exponential job. There are two minor differences with [25]: $f_*^{(p)}(t)$ and $f_\ell^{(p)}(t)$ are there denoted as $f_0(t)$ and $f_{\ell+1}(t)$ and the mean service time of a job is assumed to be 1. Note that the latter assumption can be made without loss of generality by rescaling time.

Hence, due to [25], we have for $\ell \geq 0$,

$$\begin{aligned}\frac{d}{dt}f_\ell^{(p)}(t) &= \lambda f_{\ell-1}^{(p)}(t)1[\ell \geq 1] - \lambda f_\ell^{(p)}(t) + \lambda f_*^{(p)}(t)\alpha 1[\ell = 0] + f_\ell^{(p)}(t)S^{(p)} + f_{\ell+1}^{(p)}(t)\mu\alpha \\ &\quad + r f_*^{(p)}(t)f_{\ell+1}^{(p)}(t) - r f_*^{(p)}(t)f_\ell^{(p)}(t)1[\ell \geq 1] + r f_*^{(p)}(t)\left(1 - f_*^{(p)}(t) - f_0^{(p)}(t)e\right)\alpha 1[\ell = 0],\end{aligned} \tag{5}$$

and

$$\frac{d}{dt}f_*^{(p)}(t) = -\lambda f_*^{(p)}(t) + f_0^{(p)}(t)\mu - r f_*^{(p)}(t)\left(1 - f_*^{(p)}(t) - f_0^{(p)}(t)e\right), \tag{6}$$

with

$$S^{(p)} = \begin{bmatrix} S_{00}^{(p)} & 0 \\ S_{10} & S_{11}^{(p)} \end{bmatrix}, \quad S_{00}^{(p)} = \begin{bmatrix} -\mu_2 & & & \\ \mu_2 & -\mu_2 & & \\ & \mu_2 & -\mu_2 & \\ & & \ddots & \end{bmatrix}, \quad S_{11}^{(p)} = -\mu_1 I. \tag{7}$$

As with the definition of $S^{(c)}(r,t)$, the matrix $S^{(p)}$ corresponds to events that do not lead to a change in the value of $\ell \geq 1$. In contrast to the child job stealing strategy, where $S^{(c)}(r,t)$ depended on r and t, $S^{(p)}$ is independent of r and t as any steal event changes the value of ℓ.

Note that in case of parent stealing

$$\sum_{\ell' \geq 1} f_{\ell'}^{(p)}(t)e = 1 - f_*^{(p)}(t) - f_0^{(p)}(t)e,$$

equals the probability that a probe transmitted at time t succeeds in stealing a parent job (as a parent job in service is not stolen).

5 QBD DESCRIPTION

The sets of ODEs given by (2)-(3) and (5)-(6) describe the transient evolution of the infinite system for the child and parent stealing models, respectively. We now introduce two Quasi-Birth-Death (QBD) Markov chains and show further on that their unique stationary distribution corresponds to the unique fixed points of these two mean field models. Proving this is non-trivial and is the main technical contribution of the paper.

For $i \in \{(c),(p)\}$, we define the QBD process $\{X_t^i(r), Y_t^i(r)\, Z_t^i(r) : t \geq 0\}$, where the *level* is given by X^i and the *phase* is given by (Y^i, Z^i) with generator $Q^i(r)$. Denote by $X^i \geq 0$ the number of parent jobs waiting, $Y^i \in \{0,1,\ldots,m\}$ the number of child jobs in the queue and $Z^i = \{0,1\}$ where $Z^i = 1$ if a parent is currently in service and $Z^i = 0$ if not. Define

$$\pi_*^i(r) = \lim_{t\to\infty} P[X_t^i(r) = 0, Y_t^i(r) = 0, Z_t^i(r) = 0], \tag{8}$$

and for $\ell \geq 0$,

$$\pi^i_\ell(r) = \left(\pi^i_{\ell,1,0}(r), ..., \pi^i_{\ell,m,0}(r), \pi^i_{\ell,0,1}(r), ..., \pi^i_{\ell,m,1}(r)\right), \tag{9}$$

where

$$\pi^i_{\ell,j,k}(r) = \lim_{t\to\infty} P[X^i_t(r) = \ell, Y^i_t(r) = j, Z^i_t(r) = k]. \tag{10}$$

5.1 Child job stealing

The QBD for the child stealing model is very similar to a simple M/PH/1 queue with arrival rate λ and phase-type service time characterized by $(\alpha, S^{(c)}(r))$, except that we also have additional job arrivals at some rate $\lambda_c(r)$ (defined later) when the server is idle and these additional arrivals have an exponential service time with parameter μ_2. The subgenerator matrix $S^{(c)}(r)$ is identical to $S^{(c)}(r,t)$ defined by (4), if we replace $f_*(t)$ by $q = 1-\rho$ with

$$\rho = \lambda\left(\frac{1}{\mu_1} + \frac{\sum_{n=1}^m np_n}{\mu_2}\right). \tag{11}$$

Proposition 5.1. *The mean $m_{PH} = \alpha(-S^{(c)}(r))^{-1}e$ of the phase-type distribution characterized by $(\alpha, S^{(c)}(r))$ can be written as*

$$m_{PH} = \frac{\rho}{\lambda} - \frac{1}{\mu_2}\left[\sum_{j=1}^m \tilde{p}_j\left(\frac{rq}{rq+\mu_1}\right)^j + \frac{rq}{rq+\mu_2}\sum_{j=2}^m \tilde{p}_j\left(1-\left(\frac{rq}{rq+\mu_1}\right)^{j-1}\right)\right], \tag{12}$$

where $\tilde{p}_j = \sum_{n\geq j} p_n$.

Proof. Using blockwise inversion, $(-S^{(c)}(r))^{-1}$ equals

$$\begin{bmatrix} (-S^{(c)}_{00}(r))^{-1} & 0 \\ (-S^{(c)}_{11}(r))^{-1}S_{10}(-S^{(c)}_{00}(r))^{-1} & (-S^{(c)}_{11}(r))^{-1} \end{bmatrix}, \tag{13}$$

where

$$S^{(c)}_{00}(r) = \begin{bmatrix} -\mu_2 & & \\ \mu_2+rq & -\mu_2-rq & \\ & \ddots & \ddots \end{bmatrix}, \quad S^{(c)}_{11}(r) = \begin{bmatrix} -\mu_1 & & \\ rq & -\mu_1-rq & \\ & \ddots & \ddots \end{bmatrix}. \tag{14}$$

For $i \in \mathbb{N}$, define $d_i = \frac{(rq)^i}{(\mu_1+rq)^{i+1}}$. We then have

$$(-S^{(c)}_{00}(r))^{-1} = \begin{bmatrix} \frac{1}{\mu_2} & & & \\ \vdots & \frac{1}{\mu_2+rq} & & \\ \vdots & \vdots & \ddots & \\ \frac{1}{\mu_2} & \frac{1}{\mu_2+rq} & \cdots & \frac{1}{\mu_2+rq} \end{bmatrix}, \quad (-S^{(c)}_{11}(r))^{-1} = \begin{bmatrix} \frac{\mu_1+rq}{\mu_1}d_0 & & & \\ \frac{\mu_1+rq}{\mu_1}d_1 & d_0 & & \\ \frac{\mu_1+rq}{\mu_1}d_2 & d_1 & d_0 & \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}, \tag{15}$$

and thus

$$(-S^{(c)}_{11}(r))^{-1}S_{10}(-S^{(c)}_{00}(r))^{-1} = \mu_1\begin{bmatrix} 0 & 0 & \cdots & 0 \\ \frac{1}{\mu_2}d_0 & 0 & & \vdots \\ \frac{1}{\mu_2}(d_0+d_1) & \frac{1}{\mu_2+rq}d_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \frac{1}{\mu_2}\sum_{k=0}^{m-1} d_k & \frac{1}{\mu_2+rq}\sum_{k=0}^{m-2} d_k & \cdots & \frac{1}{\mu_2+rq}d_0 \end{bmatrix}. \tag{16}$$

Table 1. Transitions for the QBDs in Section 5

$i \in \{(c),(p)\}$	From	Rate	For
1. $(c),(p)$	$(X^i, Y^i, Z^i) \to (X^i, Y^i + j, Z^i + 1)$	λp_j	$X^i = 0, Y^i = 0, Z^i = 0, j = 0, 1, \ldots m,$
2. $(c),(p)$	$(X^i, Y^i, Z^i) \to (X^i + 1, Y^i, Z^i)$	λ	$X^i \geq 0, Y^i \geq 1, Z^i = 0$ or $X^i \geq 0, Y^i \geq 0, Z^i = 1,$
3. $(c),(p)$	$(X^i, Y^i, Z^i) \to (X^i, Y^i, Z^i - 1)$	μ_1	$X^i \geq 0, Y^i \geq 1, Z^i = 1,$ or $X^i = 0, Y^i = 0, Z^i = 1,$
4. $(c),(p)$	$(X^i, Y^i, Z^i) \to (X^i, Y^i - 1, Z^i)$	μ_2	$X^i \geq 0, Y^i \geq 2, Z^i = 0,$ or $X^i = 0, Y^i = 1, Z^i = 0,$
5. $(c),(p)$	$(X^i, Y^i, Z^i) \to (X^i - 1, Y^i - 1 + j, Z^i + 1)$	$\mu_2 p_j$	$X^i \geq 1, Y^i = 1, Z^i = 0, j = 0, 1, \ldots m,$
6. $(c),(p)$	$(X^i, Y^i, Z^i) \to (X^i - 1, Y^i + j, Z^i)$	$\mu_1 p_j$	$X^i \geq 1, Y^i = 0, Z^i = 1, j = 0, 1, \ldots m,$
7. (c)	$(X^i, Y^i, Z^i) \to (X^i, Y^i + 1, Z^i)$	$\lambda_c(r)$	$X^i = 0, Y^i = 0, Z^i = 0,$
8. (c)	$(X^i, Y^i, Z^i) \to (X^i, Y^i - 1, Z^i)$	rq	$X^{(c)} \geq 0, Y^i \geq 2, Z^i = 0,$ or $X^i \geq 0, Y^i \geq 1, Z^i = 1.$
9. (p)	$(X^i, Y^i, Z^i) \to (X^i, Y^i + j, Z^i + 1)$	$\lambda_p(r) p_j$	$X^i = 0, Y^i = 0, Z^i = 0, j = 0, 1, \ldots m,$
10. (p)	$(X^i, Y^i, Z^i) \to (X^i - 1, Y^i, Z^i)$	rq	$X^{(c)} \geq 1$

Using the identity that $\mu_1 \sum_{k=0}^{s} d_k = 1 - \left(\frac{rq}{rq+\mu_1}\right)^{s+1}$, we find that $(-S_{11}^{(c)}(r))^{-1} e = e/\mu_1$ and

$$(-S_{11}^{(c)}(r))^{-1} S_{10} (-S_{00}^{(c)}(r))^{-1} e = \begin{bmatrix} 0 \\ \frac{1}{\mu_2}\left(1 - \frac{rq}{rq+\mu_1}\right) \\ \frac{1}{\mu_2}\left(1 - \left(\frac{rq}{rq+\mu_1}\right)^2\right) + \frac{1}{\mu_2+rq}\left(1 - \frac{rq}{rq+\mu_1}\right) \\ \frac{1}{\mu_2}\left(1 - \left(\frac{rq}{rq+\mu_1}\right)^3\right) + \frac{1}{\mu_2+rq}\sum_{j=1}^{2}\left(1 - \left(\frac{rq}{rq+\mu_1}\right)^j\right) \\ \vdots \\ \frac{1}{\mu_2}\left(1 - \left(\frac{rq}{rq+\mu_1}\right)^m\right) + \frac{1}{\mu_2+rq}\sum_{j=1}^{m-1}\left(1 - \left(\frac{rq}{rq+\mu_1}\right)^j\right) \end{bmatrix}.$$

As $1/(rq+\mu_2) = \frac{1}{\mu_2}\left(1 - \frac{rq}{rq+\mu_2}\right)$, the $(n+1)$-st entry of the vector $(-S_{11}^{(c)}(r))^{-1} S_{10} (-S_{00}^{(c)}(r))^{-1} e$ can be written as

$$\frac{1}{\mu_2}\left(n - \left(\frac{rq}{rq+\mu_1}\right)^n - (n-1)\frac{rq}{\mu_2+rq} - \left(1 - \frac{rq}{\mu_2+rq}\right)\sum_{j=1}^{n-1}\left(\frac{rq}{rq+\mu_1}\right)^j\right)$$
$$= \frac{1}{\mu_2}\left(n - \sum_{j=1}^{n}\left(\frac{rq}{rq+\mu_1}\right)^j - \frac{rq}{\mu_2+rq}\sum_{j=1}^{n-1}\left(1 - \left(\frac{rq}{rq+\mu_1}\right)^j\right)\right).$$

We may therefore conclude that the mean $\alpha(-S^{(c)}(r))^{-1} e$ equals

$$\frac{1}{\mu_1} + \frac{1}{\mu_2}\sum_{n=1}^{m} n p_n - \frac{1}{\mu_2}\sum_{n=1}^{m} p_n \sum_{j=1}^{n}\left(\frac{rq}{rq+\mu_1}\right)^j - \frac{1}{\mu_2}\frac{rq}{\mu_2+rq}\sum_{n=2}^{m} p_n \sum_{j=2}^{n}\left(1 - \left(\frac{rq}{rq+\mu_1}\right)^{j-1}\right),$$

from which we obtain the result in (12). □

Note that the mean of the phase-type distribution $(\alpha, S^{(c)}(r))$ is upper bounded by ρ/λ (and only equal for $r = 0$). This implies that the load of the queue (when ignoring the additional arrivals when the server is idle) is upper bounded by ρ. As such it is clear that this queueing system is stable for all $r \geq 0$ if $\rho < 1$. For completeness we provide a formal proof in Proposition 5.2.

The possible transitions for this QBD for $i = (c)$ are listed in Table 1: 1. a parent job arriving at an idle queue and proceeding directly into service, where any child jobs generated join the queue, 2. a parent arriving to a non-idle queue, 3. completion of a parent in service, not succeeded by another parent job, 4. child service completion, succeeded by either another child job or no job, 5. child service completion, succeeded by a parent job that enters service and any child jobs generated join

the queue, 6. parent service completion, succeeded by a parent job that enters service and any child jobs generated join the queue, 7. arrival of a child job due to work stealing and 8. negative arrivals due to work stealing elsewhere.

The generator of the QBD Markov chain has the following form:

$$Q^{(c)}(r) = \begin{bmatrix} -\lambda_0^{(c)}(r) & \lambda_c(r)e_1 + \lambda\alpha & & \\ \mu & A_0^{(c)}(r) & A_1 & \\ & A_{-1}^{(c)} & A_0^{(c)}(r) & A_1 \\ & & \ddots & \ddots \end{bmatrix}, \tag{17}$$

with $\lambda_0^{(c)}(r) = \lambda_c(r) + \lambda$. The size $2m + 1$ matrix $A_0^{(c)}(r)$ contains the transitions between states belonging to the same level and is given by

$$A_0^{(c)}(r) = S^{(c)}(r) - \lambda I, \tag{18}$$

The matrices $A_{-1}^{(c)}(r)$ and A_1 record the transitions for which the level is decreased and increased by one, respectively. We have

$$A_{-1}^{(c)} = \mu\alpha, \tag{19}$$

and

$$A_1 = \lambda I. \tag{20}$$

Denote by $A^{(c)}(r) = A_{-1}^{(c)} + A_0^{(c)}(r) + A_1$, the generator of the phase process, then

$$A^{(c)}(r) = S^{(c)}(r) + \mu\alpha. \tag{21}$$

The phase captures the mixture of the number of child jobs present in the queue and the type of job in service, thus the physical interpretation of this generator describes the changes due to the completion of the current job in service or when a child job is stolen which can only occur when $j + k > 1$.

Due to the QBD structure [18], we have

$$\pi_0^{(c)}(r) = \pi_*^{(c)}(r)R_0^{(c)}(r) \tag{22}$$

and for $\ell \geq 1$,

$$\pi_\ell^{(c)}(r) = \pi_0^{(c)}(r)R^{(c)}(r)^\ell, \tag{23}$$

where $R^{(c)}(r)$ is a $(2m+1)\times(2m+1)$ matrix and by [14, Proposition 6.4.2] the smallest nonnegative solution to

$$A_1 + R^{(c)}(r)A_0^{(c)}(r) + R^{(c)}(r)^2A_{-1}^{(c)} = 0. \tag{24}$$

Also,

$$\lambda_c(r)e_1 + \lambda\alpha + R_0^{(c)}(r)A_0^{(c)}(r) + R_0^{(c)}(r)R^{(c)}(r)A_{-1}^{(c)} = 0 \tag{25}$$

and

$$A_1G^{(c)}(r) = R^{(c)}(r)A_{-1}^{(c)}, \tag{26}$$

where $G^{(c)}(r)$ is the smallest nonnegative solution to

$$A_{-1}^{(c)} + A_0^{(c)}(r)G^{(c)}(r) + A_1G^{(c)}(r)^2 = 0. \tag{27}$$

Then

$$R_0^{(c)}(r) = -\left(\lambda_c(r)e_1 + \lambda\alpha\right)\left(A_0^{(c)}(r) + \lambda G^{(c)}(r)\right)^{-1}, \tag{28}$$

where $\left(A_0(r) + \lambda G^{(c)}(r)\right)$ is a subgenerator[1] matrix and is therefore invertible. We note that $R^{(c)}(r)$ and $G^{(c)}(r)$ are independent of $\lambda_c(r)$.

The physical interpretation of matrix $G^{(c)}(r)$ is that the $(i,j)^{th}$ entry of the matrix $G^{(c)}(r)$ is the probability that the QBD will first enter level $\ell - 1$ in phase j, given that it starts in phase i of level ℓ. Due to this interpretation we have

$$G^{(c)}(r) = G^{(c)} = e\alpha. \tag{29}$$

This result also follows from the structure of $A_{-1}(r)$ [14, Theorem 8.5.1] and yields an explicit formula for $\pi_0^{(c)}(r)$.

To fully characterize the QBD in terms of λ, μ_1, μ_2 and the probabilities p_i, we still need to specify $\lambda_c(r)$. The steal rate $\lambda_c(r)$ is defined as

$$\lambda_c(r) = \frac{\lambda}{q}\left[\sum_{j=1}^{m}\tilde{p}_j\left(\frac{rq}{rq+\mu_1}\right)^j + \frac{rq}{rq+\mu_2}\sum_{j=2}^{m}\tilde{p}_j\left(1-\left(\frac{rq}{rq+\mu_1}\right)^{j-1}\right)\right], \tag{30}$$

where $\tilde{p}_j = \sum_{i\geq j} p_i$ is the probability that there are j jobs to be stolen. Note that the expression between brackets is identical to the expression appearing in (12). Using probabilistic arguments one finds that the first sum in this expression corresponds to the mean number of child jobs stolen during the service of a parent job, while the second term is the mean number of child jobs that is stolen while a child is in service. The second expression relies on the fact that the number of child jobs stolen after the parent finishes its service has a binomial distribution with parameters $(k-1, rq/(rq+\mu_2))$ if there were k child jobs left when the service of the parent ended.

Proposition 5.2. *The QBD process* $\{X_t^{(c)}(r), Y_t^{(c)}(r), Z_t^{(c)}(r) : t \geq 0\}$ *has a unique stationary distribution for any* $r \geq 0$ *if* $\rho < 1$.

Proof. If suffices to check the drift condition for QBD processes [18], which states that the process is positive recurrent if $\theta^{(r)}A_{-1}^{(c)}(r)e > \theta^{(r)}A_1 e$, where $\theta^{(r)}$ is the such that $\theta^{(r)}A^{(c)}(r) = 0$, where $A^{(c)}(r)$ is defined in (21). Denote

$$\theta^{(r)} = (\theta^{(r)}_{(0,1)}, \ldots, \theta^{(r)}_{(0,m)}, \theta^{(r)}_{(1,0)}, \theta^{(r)}_{(1,1)}, \ldots, \theta^{(r)}_{(1,m)}), \tag{31}$$

then

$$\theta^{(r)}_{(0,1)} = \frac{1}{\mu_2}\sum_{j=1}^{m} p_j\left(1-\left(\frac{rq}{rq+\mu_1}\right)^j\right), \tag{32}$$

$$\theta^{(r)}_{(0,i)} = \frac{1}{rq+\mu_2}\sum_{j=i}^{m} p_j\left(1-\left(\frac{rq}{rq+\mu_1}\right)^{j-i+1}\right), \tag{33}$$

for $i = 2, \ldots, m$ and

$$\theta^{(r)}_{(1,0)} = \frac{1}{\mu_1}\sum_{j=0}^{m} p_j\left(\frac{rq}{rq+\mu_1}\right)^j, \tag{34}$$

[1] A matrix is a subgenerator if its diagonal entries are negative, its off-diagonal entries are non-negative and its row sums are negative.

$$\theta^{(r)}_{(1,i')} = \frac{1}{rq+\mu_1}\sum_{j=i'}^{m} p_j \left(\frac{rq}{rq+\mu_1}\right)^{j-i'}, \tag{35}$$

for $i' = 1, \ldots, m$. It can be readily verified that $\theta^{(r)}A^{(c)}(r) = 0$. Using these expressions one finds that

$$\theta^{(r)}A^{(c)}_{-1}e = \theta^{(r)}\mu = 1.$$

By (34) and (35) we find $\sum_{i=0}^{m}\theta^{(r)}_{(1,i)} = \frac{1}{\mu_1}$, while combining (32) and (33) yields

$$\sum_{i=1}^{m}\theta^{(r)}_{(0,i)} = \frac{1}{rq+\mu_2}\sum_{j=2}^{m} jp_j + \frac{1}{\mu_2}p_1\left(1-\frac{rq}{rq+\mu_1}\right) - \frac{1}{\mu_1}\underbrace{\left(\frac{rq+\mu_1}{rq+\mu_2}-\frac{\mu_1}{\mu_2}\right)}_{\geq 0}\sum_{j=2}^{m} p_j\left(1-\left(\frac{rq}{rq+\mu_1}\right)^j\right)$$

$$\leq \frac{1}{\mu_2}\left(\sum_{j=2}^{m} jp_j\right) + \frac{1}{\mu_2}p_1.$$

As $A_1 = \lambda I$, this shows that the upward drift $\theta^{(r)}A_1 e$ is at most ρ. □

Proposition 5.3. *We have* $\pi^{(c)}_*(r) = q$.

Proof. Due to (22) and (23) we have

$$\pi^{(c)}_*(r) = \frac{1}{1+\sum_{\ell\geq 0} R^{(c)}_0(r)(R^{(c)}(r))^\ell e}. \tag{36}$$

By Proposition 5.2 and [14, Proposition 6.4.2], we have the spectral radius of $R^{(c)}(r)$ less than one and $R^{(c)}(r) = A_1(-U(r))^{-1}$ with $U(r) = A^{(c)}_0(r) + A_1G^{(c)}$. Therefore

$$\begin{aligned}\sum_{\ell\geq 0} R^{(c)}_0(r)(R^{(c)}(r))^\ell &= (\lambda_c(r)e_1+\lambda\alpha)(-U(r))^{-1}\sum_{\ell\geq 0}\lambda^\ell(-U(r))^{-\ell},\\ &= \frac{\lambda_c(r)e_1+\lambda\alpha}{\lambda}\sum_{\ell\geq 0}\lambda^\ell(-U(r))^{-\ell} - \frac{\lambda_c(r)e_1+\lambda\alpha}{\lambda},\\ &= \frac{\lambda_c(r)e_1+\lambda\alpha}{\lambda}(I+\lambda U(r)^{-1})^{-1} - \frac{\lambda_c(r)e_1+\lambda\alpha}{\lambda},\end{aligned} \tag{37}$$

where $U(r) = A^{(c)}_0(r) + \lambda e\alpha$. Using the Woodbury matrix identity [10] we get:

$$(I+\lambda U(r)^{-1})^{-1} = I - \lambda(U(r)+\lambda I)^{-1}, \tag{38}$$

and thus:

$$\sum_{\ell\geq 0} R^{(c)}_0(r)(R^{(c)}(r))^\ell = -(\lambda_c(r)e_1+\lambda\alpha)(U(r)+\lambda I)^{-1}. \tag{39}$$

Employing the Sherman–Woodbury formula [10] and the fact that $U(r)+\lambda I = S^{(c)}(r) + \lambda e\alpha$, we further have

$$-(U(r)+\lambda I)^{-1} = (-S^{(c)}(r))^{-1} + \frac{\lambda(-S^{(c)}(r))^{-1}e\alpha(-S^{(c)}(r))^{-1}}{1-\lambda\alpha(-S^{(c)}(r))^{-1}e}.$$

Letting $m_{PH} = \alpha(S^{(c)}(r))^{-1}e$, this implies

$$-\alpha(U(r)+\lambda I)^{-1}e = m_{PH} + \frac{\lambda m^2_{PH}}{1-\lambda m_{PH}} = \frac{m_{PH}}{1-\lambda m_{PH}}, \tag{40}$$

$$-e_1(U(r)+\lambda I)^{-1}e = \frac{1}{\mu_2} + \frac{1}{\mu_2}\frac{\lambda m_{PH}}{1-\lambda m_{PH}}, \tag{41}$$

as $e_1(-S^{(c)}(r))^{-1}e = 1/\mu_2$. Combining (39), (40) and (41) yields

$$\begin{aligned}\sum_{\ell\geq 0} R_0^{(c)}(r)(R^{(c)}(r))^{\ell}e &= \frac{\lambda_c(r)}{\mu_2}\left(1+\frac{\lambda m_{PH}}{1-\lambda m_{PH}}\right)+\frac{\lambda m_{PH}}{1-\lambda m_{PH}},\\ &= \frac{\lambda}{q}\left(\frac{\rho}{\lambda}-m_{PH}\right)\left(1+\frac{\lambda m_{PH}}{1-\lambda m_{PH}}\right)+\frac{\lambda m_{PH}}{1-\lambda m_{PH}},\\ &= \frac{1-q}{q},\end{aligned}$$

where the second equality follows from (30) and (12). The result now follows from (36). □

5.2 Parent job stealing

The QBD process for the system with parent job stealing is a special case of the one used in [25]. Compared to the QBD for the system with child job stealing, this queue is not similar to an M/PH/1 queue. Instead it corresponds to an M/PH/1 queue subject to negative arrivals (when the queue has pending jobs) and these correspond to parent jobs that are stolen. The possible transitions for this QBD for $i = (p)$ are listed in Table 1.

The generator of the process $\{X_t^{(p)}(r), Y_t^{(p)}(r), Z_t^{(p)}(r)\}$ is

$$Q^{(p)}(r) = \begin{bmatrix} -\lambda_0^{(p)}(r) & (\lambda+\lambda_p(r))\alpha & & \\ \mu & B_0^{(p)} & A_1 & \\ & A_{-1}^{(p)}(r) & A_0^{(p)}(r) & A_1 \\ & & \ddots & \ddots \end{bmatrix}, \tag{42}$$

with $\lambda_0^{(p)}(r) = \lambda + \lambda_p(r)$,

$$A_{-1}^{(p)}(r) = A_{-1}^{(c)} + rqI, \tag{43}$$

where $A_{-1}^{(c)}$ is given in Equation (19) and

$$A_0^{(p)}(r) = S^{(p)} - \lambda I - rqI, \tag{44}$$

where $S^{(p)}$ is given by (7) and

$$B_0^{(p)} = S^{(p)} - \lambda I. \tag{45}$$

We have

$$\pi_0^{(p)}(r) = \pi_*^{(p)}(r)R_0^{(p)}(r), \tag{46}$$

and for $\ell \geq 1$,

$$\pi_\ell^{(p)}(r) = \pi_0^{(p)}(r)R^{(p)}(r)^{\ell}, \tag{47}$$

where

$$R_0^{(p)}(r) = -(\lambda+\lambda_p(r))\alpha\left(B_0^{(p)} + A_1G^{(p)}(r)\right)^{-1}, \tag{48}$$

and $R^{(p)}(r)$ is the smallest nonnegative solution to

$$A_1 + R^{(p)}(r)A_0^{(p)}(r) + R^{(p)}(r)^2A_{-1}^{(p)} = 0. \tag{49}$$

The value of $\lambda_p(r)$ is determined by demanding that $\pi_*^{(p)}(r) = q = 1 - \rho$. This is in contrast to the child stealing scenario in which we gave an explicit expression for λ_c and showed in Proposition 5.3 that $\pi_*^{(c)}(r) = q$. As indicated in [25], this yields

$$\lambda_0^{(p)}(r) = \frac{\rho}{q\alpha(\lambda(I - G^{(p)}(r)) - S^{(p)})^{-1}(I - R^{(p)}(r))^{-1}e} - \lambda, \tag{50}$$

while the steady state probabilities of this system are given by

$$\pi_*^{(p)}(r) = q, \tag{51}$$

$$\pi_\ell^{(p)}(r) = \rho\frac{\alpha(\lambda(I - G^{(p)}(r)) - S^{(p)})^{-1}R^{(p)}(r)^\ell}{\alpha(\lambda(I - G^{(p)}(r)) - S^{(p)})^{-1}(I - R^{(p)}(r))^{-1}e}, \tag{52}$$

for $\ell \geq 0$. Contrary to the case with child job stealing, the matrices $R^{(p)}(r)$ and $G^{(p)}(r)$ must be determined numerically by solving a non-linear matrix equation, which can be computed using the cyclic or logarithmic reduction algorithms with quadratic convergence [1].

6 RESPONSE TIME DISTRIBUTION OF THE QBD

Define $T^i(r)$ as the response time for a job with probe rate r, in a system where only type $i \in \{(p), (c)\}$ jobs can be transferred. The response time is the interval of time between the arrival epoch of a parent job and the instant at which the parent and all of its child jobs have completed service. This can be expressed as

$$T^i(r) = W^i(r) + J^i(r), \tag{53}$$

where $W^i(r)$ is waiting time defined as the interval between the arrival epoch of a parent job and instant at which it moves into service. In the case that parent jobs are stolen, we assume that the oldest waiting parent job is stolen (as this should be best to reduce the variability of the waiting time). The service time $J^i(r)$ is defined as the time between the start of service of the parent job and the first point in time in which both the parent and all of its child jobs have completed service. Clearly $W^i(r)$ and $J^i(r)$ are independent. Note that $W^i(r)$ is harder to compute in case parent jobs are stolen, while $J^i(r)$ is more demanding when child jobs can be transferred.

6.1 Waiting time distribution

We present a unified analysis for both models. Due to the PASTA property we have $P[W^i(r) = 0] = q$. To compute $P[W^i(r) > t]$ we employ the approach taken in Ozawa [19] and Horvath *et al.* [12]. Ozawa [19] studied FIFO queues defined by a QBD Markov chain where transitions that increase/decrease the level are regarded as arrivals/departures. Ozawa showed that the sojourn time distribution (the time between an arrival and its departure) of a queue defined by a QBD has a matrix exponential form (of order n^2 if we have n phases per level). A similar result is presented below for the waiting time $W^i(r)$.

Theorem 6.1. *For $i \in \{(p), (c)\}$, the distribution of the waiting time is given by*

$$P[W^i(r) > t] = (e' \otimes \pi_0^i(I - R^i(r))^{-1})e^{\mathbb{W}^i t}vec\langle I\rangle, \tag{54}$$

with $\mathbb{W}^i = ((A_0^i(r) + A_1)' \otimes I) + ((A_{-1}^i(r))' \otimes R^i(r))$, where $\otimes$ denotes the Kronecker product and where $vec\langle\cdot\rangle$ is the column stacking operator, i.e., $vec\langle I\rangle$ is the vector obtained by stacking the columns of I. The mean waiting time is

$$E\left[W^i(r)\right] = \int_0^\infty P\left[W^i(r) > t\right] dt = (e' \otimes \pi_0^i(I - R^i(r))^{-1})(-\mathbb{W}^i)^{-1}vec\langle I\rangle. \tag{55}$$

Table 2. Non-zero entries of the rate matrix $\tilde{S}^{(c)}(r)$

	From	Rate	For
1.	$(Y^{(c)}, Z^{(c)}, \tilde{Y}^{(c)}) \to (Y^{(c)} - 1, Z^{(c)}, \tilde{Y}^{(c)})$	μ_2	$Y^{(c)} \geq 1, Z^{(c)} = 0$
2.	$(Y^{(c)}, Z^{(c)}, \tilde{Y}^{(c)}) \to (Y^{(c)}, Z^{(c)} - 1, \tilde{Y}^{(c)})$	μ_1	$Z^{(c)} = 1$
3.	$(Y^{(c)}, Z^{(c)}, \tilde{Y}^{(c)}) \to (Y^{(c)}, Z^{(c)}, \tilde{Y}^{(c)} - 1)$	$\mu_2 \tilde{Y}^{(c)}$	$\tilde{Y}^{(c)} \geq 1$
4.	$(Y^{(c)}, Z^{(c)}, \tilde{Y}^{(c)}) \to (Y^{(c)} - 1, Z^{(c)}, \tilde{Y}^{(c)} + 1)$	$rq^{(c)}$	$Y^{(c)} + Z^{(c)} \geq 2$

Proof. Let $(N^i(k,t))_{j,j'}$ be the probability that we have exactly k transitions that decrease the level by one in $(0,t)$ and the phase at time t equals j' for the QBD Q^i given that the level never decreased below 1 and the phase was j at time 0. Due to the PASTA property we have

$$P[W^i(r) > t] = \sum_{n=1}^{\infty} \pi^i_{n-1} \sum_{k=0}^{n-1} N^i(k,t)e,$$

as $(\pi^i_{n-1})_j$ is the probability that a tagged parent job is the n^{th} parent job waiting in the queue immediately after it arrived and the service phase equals j. In such case there can be at most $n-1$ events that decrease the level otherwise $W^i(r) < t$. Thus,

$$P[W^i(r) > t] = \sum_{k=0}^{\infty} \pi^i_0 \sum_{n=k+1}^{\infty} (R^i(r))^{n-1} N^i(k,t)e = \pi^i_0 (I - R^i(r))^{-1} \sum_{k=0}^{\infty} (R^i(r))^k N^i(k,t)e.$$

Using the same arguments as in [19] or [12] one finds that

$$vec\left\langle \sum_{k=0}^{\infty} (R^i(r))^k N^i(k,t) \right\rangle = e^{\mathbb{W}^i t} vec\,\langle I \rangle\,.$$

The proof is completed by noting that $vec\langle ABC\rangle = (C' \otimes A) vec\langle B\rangle$.

□

6.2 Service distribution

When parent jobs are stolen, a parent job and all its child jobs are executed on the same server. Hence, the service time $J^{(p)}$ has a phase type distribution with parameters $(\alpha, S^{(p)})$:

$$P[J^{(p)} < t] = 1 - \alpha e^{-S^{(p)}t} e, \tag{56}$$

and $E[J^{(p)}] = \alpha \left(-S^{(p)}\right)^{-1} e$. We have from (54) that the waiting time distribution $W^i(r)$ follows a matrix exponential distribution with parameters $(e' \otimes \pi^i_0 (I - R^i(r))^{-1}, \mathbb{W}^i, (-\mathbb{W}^i) vec\langle I\rangle)$. Therefore due to [2, Theorem 4.4.2] the convolution of the waiting time and service time can be expressed as

$$P[T^{(p)}(r) > t] = [(e' \otimes \pi^{(p)}_0 (I - R^{(p)}(r))^{-1})\, q\alpha] e^{\mathbb{T}^p t} (-\mathbb{T}^p)^{-1} \begin{pmatrix} 0_{2m+1} \\ \mu \end{pmatrix}, \tag{57}$$

where

$$\mathbb{T}^c = \begin{bmatrix} \mathbb{W}^c & (-\mathbb{W}^c) vec\langle I\rangle \tilde{\alpha} \\ 0 & \tilde{S}^{(c)} \end{bmatrix},$$

and

$$E[T^{(p)}(r) > t] = [(e' \otimes \pi^{(p)}_0 (I - R^{(p)}(r))^{-1})] (\mathbb{T}^p)^{-2} \begin{pmatrix} 0_{2m+1} \\ \mu \end{pmatrix}. \tag{58}$$

When child jobs are stolen, we need to keep track of the number of transferred child jobs as such a child job may be the last to complete service. To this end, we define the phase process $\{Y_t^{(c)}(r), Z_t^{(c)}(r), \tilde{Y}_t^{(c)}(r)\}_{t\geq 0}$ where $Y_t^{(c)}(r), Z_t^{(c)}(r)$ are defined as before and $\tilde{Y}_t^{(c)} \in \{0, 1, \ldots, m\}$ is the number of transferred child jobs still in service.

The service time $J^{(c)}(r)$ of a parent job with n child jobs therefore equals the time needed for the phase process to go from phase $(n, 1, 0)$ to $(0, 0, 0)$. In other words $J^{(c)}(r)$ can be represented as a phase-type distribution with parameters $(\tilde{\alpha}^{(c)}, \tilde{S}^{(c)}(r))$. As $Z_t^{(c)}(r) \in \{0, 1\}$, $0 \leq \tilde{Y}_t^{(c)}(r) + Y_t^{(c)}(r) \leq m$ and $(0, 0, 0)$ is the absorbing state $\tilde{S}^{(c)}(r)$ is a size $d = 2\sum_{k=1}^{m+1} k - 1 = m^2 + 3m + 1$ matrix. Its non-zero entries are listed in Table 2. The vector $\tilde{\alpha}^{(c)}$ equals p_n in the position corresponding to phase $(n, 1, 0)$ and equals zero for any other phase. Then

$$P[J^{(c)}(r) < t] = 1 - \tilde{\alpha}^{(c)} e^{-\tilde{S}^{(c)}(r)t} e, \tag{59}$$

and $E[J^{(c)}(r)] = \tilde{\alpha}^{(c)}(-\tilde{S}^{(c)}(r))^{-1}e$. The convolution of $W^c(r)$ and $J^c(r)$ can be computed in a similar manner as in the parent job stealing case.

7 STATIONARY BEHAVIOUR

In this section we show that the stationary distribution of the child stealing QBD in Section 5 corresponds to the unique fixed point ζ^i of the set of ODEs. For the case of parent job stealing, we illustrate how the results by [25] can be modified to obtain the desired result. Define for $i \in \{(c), (p)\}$, $\zeta^i = (\zeta_*^i, \zeta_0^i, \zeta_1^i, \ldots)$ with $\zeta_*^i + \sum_{\ell\geq 0} \zeta_\ell^i e = 1$.

7.1 Child job stealing

Lemma 7.1. *For any fixed point $\zeta^{(c)} = (\zeta_*^{(c)}, \zeta_0^{(c)}, \zeta_1^{(c)}, \ldots)$ with $\zeta_*^{(c)} + \sum_{\ell\geq 0} \zeta_\ell^{(c)} e = 1$ of the set of ODEs in Equations (2)-(3) we have*

$$\lambda = \lambda\zeta_*^{(c)} + \sum_{\ell\geq 1} \zeta_\ell^{(c)} \mu. \tag{60}$$

Proof. As $\frac{d}{dt} f_\ell^{(c)}(t) = 0$ in a fixed point we can show that

$$\sum_{\ell\geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_m \\ \mu_1 \\ 0_m \end{pmatrix} + \sum_{\ell\geq 0} \zeta_\ell^{(c)} \begin{pmatrix} \mu_2 \\ 0_m \\ 0_m \end{pmatrix} = \lambda + r\zeta_*^{(c)} \sum_{\ell\geq 0} \zeta_\ell^{(c)} (e - v_0), \tag{61}$$

by using the equality $\sum_{\ell\geq 0}(\ell + 1)\frac{d}{dt} f_\ell^{(c)}(t)e = 0$. (60) now follows by combining this equality with $\frac{d}{dt} f_*^{(c)}(t) = 0$. □

Lemma 7.2. *For any fixed point $\zeta^{(c)} = (\zeta_*^{(c)}, \zeta_0^{(c)}, \zeta_1^{(c)}, \ldots)$ with $\zeta_*^{(c)} + \sum_{\ell\geq 0} \zeta_\ell^{(c)} e = 1$ of the set of ODEs in Equations (2)-(3) we have for $1 \leq k \leq m$*

$$r\zeta_*^{(c)} \sum_{\ell\geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_{m+k} \\ 1_{m-k+1} \end{pmatrix} = \lambda \sum_{j=k}^{m} \tilde{p}_j \left(\frac{r\zeta_*^{(c)}}{r\zeta_*^{(c)} + \mu_1} \right)^{j-k+1}, \tag{62}$$

where 1_i denotes a column vector of i ones.

Proof. We use backward induction on k to prove this result. By demanding that

$$\sum_{\ell\geq 0} \frac{d}{dt} f_\ell^{(c)}(t)[0'_{m+k}\ 1'_{m-k+1}]' = 0,$$

for any $k \in \{1, \dots, m\}$, we find due to Lemma 7.1 that

$$\lambda \tilde{p}_k = r\zeta_*^{(c)} \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_{m+k} \\ 1 \\ 0_{m-k} \end{pmatrix} + \mu_1 \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_{m+k} \\ 1_{m-k+1} \end{pmatrix}, \tag{63}$$

which is equivalent to (62) when $k = m$. For $k < m$ we can rewrite the above as

$$\lambda \tilde{p}_k = (r\zeta_*^{(c)} + \mu_1) \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_{m+k} \\ 1_{m-k+1} \end{pmatrix} - r\zeta_*^{(c)} \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_{m+k+1} \\ 1_{m-k} \end{pmatrix}, \tag{64}$$

and use induction on the second term. This yields

$$\lambda \tilde{p}_k = (r\zeta_*^{(c)} + \mu_1) \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_{m+k} \\ 1_{m-k+1} \end{pmatrix} - \lambda \sum_{j=k+1}^{m} \tilde{p}_j \left(\frac{r\zeta_*^{(c)}}{r\zeta_*^{(c)} + \mu_1} \right)^{j-k}, \tag{65}$$

from which we obtain the result in (62). □

Proposition 7.3. *For any fixed point $\zeta^{(c)} = (\zeta_*^{(c)}, \zeta_0^{(c)}, \zeta_1^{(c)}, \dots)$ with $\zeta_*^{(c)} + \sum_{\ell \geq 0} \zeta_\ell^{(c)} e = 1$ of the set of ODEs in Equations (2)-(3) we have*

$$\zeta_*^{(c)} = q, \tag{66}$$

$$r\zeta_*^{(c)} \sum_{\ell \geq 0} \zeta_\ell^{(c)} (e - v_0) = \zeta_*^{(c)} \lambda_c(r), \tag{67}$$

where $\lambda_c(r)$ was defined in (30).

Proof. We denote $1{:}i$ for the column vector of length i with the j^{th} entry equal to j, with $1 \leq j \leq i$. To establish that $\zeta_*^{(c)} = q$ it suffices to establish the following two identities:

$$\sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_m \\ 1_{m+1} \end{pmatrix} = \frac{\lambda}{\mu_1}, \tag{68}$$

$$\sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 1_m \\ 0_{m+1} \end{pmatrix} = \frac{\lambda}{\mu_2} \left(\sum_{i=1}^{m} i p_i \right). \tag{69}$$

As $\sum_{\ell \geq 0} \frac{d}{dt} f_\ell^{(c)}(t) \begin{pmatrix} 1_m \\ 0_{m+1} \end{pmatrix} = 0$, we find

$$\sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} \mu_2 \\ 0_m \\ 0_m \end{pmatrix} = r\zeta_*^{(c)} \sum_{\ell \geq 0} \zeta_\ell^{(c)} (e - v_0) + \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_{m+1} \\ \mu_1 1_m \end{pmatrix}. \tag{70}$$

Combining (61) and (70) yields (68). From $\sum_{\ell \geq 0} \frac{d}{dt} f_\ell^{(c)}(t)[(1{:}m)'\ 0\ (1{:}m)']' = 0$ and Lemma 7.1, one can show that

$$\sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} \mu_2 1_m \\ 0_{m+1} \end{pmatrix} = \lambda \left(\sum_{i=1}^{m} i p_i \right), \tag{71}$$

which is equivalent to (69).

Proving (67) requires more work. We prove the following two equalities that together provide us with the required result:

$$r\zeta_*^{(c)} \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_{m+1} \\ 1_m \end{pmatrix} = \lambda \sum_{j=1}^{m} \tilde{p}_j \left(\frac{r\zeta_*^{(c)}}{r\zeta_*^{(c)} + \mu_1} \right)^{j}, \tag{72}$$

$$(r\zeta_*^{(c)} + \mu_2) \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0 \\ 1_{m-1} \\ 0_{m+1} \end{pmatrix} = \lambda \sum_{j=2}^{m} \tilde{p}_j \left(1 - \left(\frac{r\zeta_*^{(c)}}{r\zeta_*^{(c)} + \mu_1} \right)^{j-1} \right). \tag{73}$$

The first is immediate from Lemma 7.2 if we set $k = 1$. To establish the second equality, we first note that $\sum_{\ell \geq 0} \frac{d}{dt} f_\ell^{(c)}(t)[0\ (1{:}(m-1))'\ 0'_{m+1}]' = 0$ allows us to show that

$$(r\zeta_*^{(c)} + \mu_2) \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0 \\ 1_{m-1} \\ 0_{m+1} \end{pmatrix} = \mu_1 \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0_{m+2} \\ 1{:}(m-1) \end{pmatrix}. \tag{74}$$

We have

$$\begin{pmatrix} 0_{m+2} \\ 1{:}(m-1) \end{pmatrix} = \sum_{k=2}^{m} \begin{pmatrix} 0_{m+k} \\ 1_{m-k+1} \end{pmatrix},$$

and combining this with (74) and Lemma 7.2 for $k = 2$ to m we find

$$\begin{aligned}
(r\zeta_*^{(c)} + \mu_2) \sum_{\ell \geq 0} \zeta_\ell^{(c)} \begin{pmatrix} 0 \\ 1_{m-1} \\ 0_{m+1} \end{pmatrix} &= \frac{\lambda\mu_1}{r\zeta_*^{(c)}} \sum_{k=2}^{m} \sum_{j=k}^{m} \tilde{p}_j \left(\frac{r\zeta_*^{(c)}}{r\zeta_*^{(c)} + \mu_1} \right)^{j-k+1}, \\
&= \frac{\lambda\mu_1}{r\zeta_*^{(c)}} \sum_{j=2}^{m} \tilde{p}_j \sum_{s=1}^{j-1} \left(\frac{r\zeta_*^{(c)}}{r\zeta_*^{(c)} + \mu_1} \right)^{s}, \\
&= \frac{\lambda\mu_1}{r\zeta_*^{(c)} + \mu_1} \sum_{j=2}^{m} \tilde{p}_j \frac{1 - \left(\frac{r\zeta_*^{(c)}}{r\zeta_*^{(c)} + \mu_1} \right)^{j-1}}{1 - \left(\frac{r\zeta_*^{(c)}}{r\zeta_*^{(c)} + \mu_1} \right)}, \\
&= \lambda \sum_{j=2}^{m} \tilde{p}_j \left(1 - \left(\frac{r\zeta_*^{(c)}}{r\zeta_*^{(c)} + \mu_1} \right)^{j-1} \right),
\end{aligned}$$

which proves (73). □

Theorem 7.4. *The stationary distribution $\pi^{(c)}(r)$ of the QBD Markov chain characterized by $Q^{(c)}(r)$ is the unique fixed point $\zeta^{(c)}$ of the set of ODEs in Equations (2)-(3).*

Proof. Using Proposition 7.3 we show that the fixed point equations $\frac{d}{dt} f_\ell^{(c)}(t) = 0$ are equivalent to the balance equations of the QBD Markov chain characterized by $Q^{(c)}(r)$. The uniqueness of the fixed point the follows from the uniqueness of the stationary distribution of the Markov chain.

For $\ell \geq 1$, $\frac{d}{dt} f_\ell^{(c)}(t) = 0$ can be written as

$$0 = \zeta_{\ell-1}^{(c)} (\lambda I) + \zeta_\ell^{(c)} \left(S^{(c)}(r,t) - \lambda I \right) + \zeta_{\ell+1}^{(c)} \mu\alpha,$$

which is exactly the balance equations of $Q^{(c)}(r)$ for $\ell \geq 1$ as $\zeta_*^{(c)} = q$ due to Proposition 7.3. This implies that $\zeta_\ell^{(c)} = \zeta_0^{(c)} R(r)^\ell$, for all $\ell \geq 1$ for any fixed point.

For $\ell = 0$, $\frac{d}{dt} f_\ell^{(c)}(t) = 0$ implies

$$0 = \zeta_0^{(c)} (S(t) - \lambda I) + \zeta_1 \mu\alpha + \lambda\zeta_*^{(c)} \alpha + r\zeta_*^{(c)} \sum_{\ell' \geq 0} \zeta_{\ell'}^{(c)} (e - v_0) e_1.$$

Due to Proposition 7.3 we can rewrite this as

$$0 = \zeta_0^{(c)} A_0(r) + \zeta_1^{(c)} A_{-1} + q (\lambda_c(r) e_1 + \lambda\alpha).$$

This indicates that $\frac{d}{dt} f_\ell^{(c)}(t) = 0$ corresponds to the balance equation for $\ell = 0$. Finally one readily checks that $\frac{d}{dt} f_*^{(c)}(t) = 0$ is equivalent to the first balance equation due to (67). □

7.2 Parent job stealing

THEOREM 7.5. *The stationary distribution $\pi^{(p)}(r)$ of the QBD Markov chain characterized by $Q^{(p)}(r)$ is the unique fixed point $\zeta^{(p)}$ of the set of ODEs in Equations (5)-(6).*

PROOF. Define $\theta^{(p)} = (\theta_1^{(p)}, \ldots, \theta_{2m+1}^{(p)})$ with $\theta_j^{(p)} = \frac{1}{\mu_2}\tilde{p}_j$ for $j = 1, \ldots, m$ and $\theta_j^{(p)} = \frac{1}{\mu_1} p_{j-m-1}$ for $j = m+1, \ldots, 2m+1$. We then have $\theta^{(p)}(S^{(p)} + \mu\alpha) = 0$. Therefore

$$\beta^{(p)} = \frac{1}{\sum_{j=1}^{2m+1} \theta_j^{(p)}} \theta^{(p)} = \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} \sum_{j=1}^{m} j p_j \right)^{-1} \theta^{(p)} \tag{75}$$

is the stationary distribution of the service phase given the server is busy. We also have

$$\beta^{(p)} \mu = \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} \sum_{j=1}^{m} j p_j \right)^{-1}. \tag{76}$$

One can now make the same calculations as in [25, Proposition 1] to conclude that

$$\sum_{\ell \geq 0} \zeta_\ell^{(p)} = \lambda \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} \sum_{j=1}^{m} j p_j \right) \beta^{(p)} = \rho \beta^{(p)}. \tag{77}$$

As $\beta^{(p)}$ is a stochastic vector, we get

$$\zeta_*^{(p)} = 1 - \rho \beta^{(p)} e = q. \tag{78}$$

The rest of the proof is identical to the one of [25, Theorem 1], except with q instead of $1 - \lambda$. □

8 PROBE RATE $r \to \infty$

In this section we present some explicit results for the case where the probe rate r tends to infinity.

Child job stealing. Taking $r \to \infty$, we define $\lambda_c = \lim_{r \to \infty} \lambda_c(r)$, then by Equation (30),

$$\lambda_c = \frac{\lambda}{q} \sum_{j=1}^{m} j p_j. \tag{79}$$

The resulting process is given by the QBD $\{X^{(c)}, Z^{(c)}\}$ as defined in Section 5, with $X \geq 0$ and $Z \in \{0, 1\}$, noting the empty boundary state is distinct from the state $(X^{(c)}, Z^{(c)}) = (0, 0)$, in which a child job is in service. The rate matrix is

$$Q_\infty^{(c)} = \begin{bmatrix} -\lambda - \lambda_c & \lambda_c e_1 + \lambda e_2 & & \\ \mu_\infty & A_0^\infty & A_1^\infty & \\ & A_{-1}^\infty & A_0^\infty & A_1^\infty \\ & & \ddots & \ddots \end{bmatrix}, \tag{80}$$

where $\mu_\infty = [\mu_2, \mu_1]'$, $A_{-1}^\infty = [0_2, \mu_\infty]$, $A_0^\infty = -\text{diag}([\mu_2 + \lambda, \mu_1 + \lambda])$ and $A_1^\infty = \lambda I$. Proceeding similarly to Section 5.1, we find

$$G_\infty^{(c)} = \begin{bmatrix} 0_2 & 1_2 \end{bmatrix}, \tag{81}$$

$$R_{0,\infty}^{(c)} = \begin{bmatrix} \frac{\lambda_c}{\mu_2 + \lambda} & \frac{\lambda}{\mu_1} \left(1 + \frac{\lambda_c}{\mu_2 + \lambda} \right) \end{bmatrix}. \tag{82}$$

Using [14, Proposition 6.4.2], we get

$$R_{\infty}^{(c)} = \lambda \begin{bmatrix} \frac{1}{\mu_2+\lambda} & \frac{\lambda}{\mu_1(\mu_2+\lambda)} \\ 0 & \frac{1}{\mu_1} \end{bmatrix}. \tag{83}$$

Note that $(I - R_{\infty}^{(c)})$ is invertible as $\lambda < \mu_1$ (it can be shown algebraically that $(I - R_{\infty}^{(c)})$ is singular only when $\lambda = \mu_1$). From (83), we get for $k \geq 1$

$$\pi_{k,0} = \pi_{0,0} \left(\frac{\lambda}{\mu_2 + \lambda} \right)^k. \tag{84}$$

According to the PASTA property,

$$\begin{aligned} E[W^{(c)}] &= \frac{1}{\mu_2} \sum_{k=0}^{\infty} \pi_{k,0} + \frac{1}{\mu_1} \sum_{k=0}^{\infty} \pi_{k,1} + \frac{1}{\mu_1} \sum_{k=1}^{\infty} k(\pi_{k,0} + \pi_{k,1}), \\ &= \left(\frac{1}{\mu_2} - \frac{1}{\mu_1} \right) \pi_{0,0} \sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu_2 + \lambda} \right)^k + \frac{1}{\mu_1} \sum_{k=0}^{\infty} (k+1)[\pi_{k,0}\ \pi_{k,1}]e, \\ &= \left(\frac{1}{\mu_2} - \frac{1}{\mu_1} \right) \pi_{0,0} \frac{\mu_2 + \lambda}{\mu_2} + \frac{1}{\mu_1} [\pi_{0,0}\ \pi_{0,1}] \sum_{k=0}^{\infty} (k+1) \left(R_{\infty}^{(c)} \right)^k e, \\ &= \left(\frac{1}{\mu_2} - \frac{1}{\mu_1} \right) \pi_{0,0} \frac{\mu_2 + \lambda}{\mu_2} + \frac{1}{\mu_1} [\pi_{0,0}\ \pi_{0,1}] \left(I - R_{\infty}^{(c)} \right)^{-2} e. \end{aligned}$$

With $\pi_0 = \pi_* R_{0,\infty}^{(c)} = q R_{0,\infty}^{(c)}$,

$$E[W^{(c)}] = \left(\frac{1}{\mu_2} - \frac{1}{\mu_1} \right) \frac{q\lambda_c}{\mu_2} + \frac{q}{\mu_1} R_{0,\infty}^{(c)} \left(I - R_{\infty}^{(c)} \right)^{-2} e, \tag{85}$$

where

$$R_{0,\infty}^{(c)} \left(I - R_{\infty}^{(c)} \right)^{-2} = \begin{bmatrix} \frac{\lambda_c(\mu_2+\lambda)}{(\mu_2)^2} & \frac{\lambda_c \lambda^2}{(\mu_2)^2(\mu_1-\lambda)} + \frac{\lambda_c \lambda^2 \mu_1 + \lambda \mu_2 \mu_1 (\mu_2+\lambda+\lambda_c)}{\mu_2(\mu_1-\lambda)^2(\mu_2+\lambda)} \end{bmatrix}. \tag{86}$$

By combining (85) and (86), we find

$$E[W^{(c)}] = \frac{q}{\mu_2^2(\lambda - \mu_1)^2} (\lambda_c \mu_1^2 + \lambda \lambda_c (\mu_2 - \mu_1) + \lambda \mu_2^2).$$

Using (79) with $\tilde{p} = \sum_j j p_j$ and $q = 1 - \rho$, this yields that the limiting waiting time is given by

$$E[W^{(c)}] = \frac{\lambda \left(\frac{1}{\mu_1} + \tilde{p} \frac{\mu_1}{\mu_2^2} \right)}{\mu_1 - \lambda}. \tag{87}$$

It is worth noting that this limit depends only on the distribution $(p_1, \ldots, p_m)$ of the number of child jobs via its mean of $\check{p}$. In other words, as r becomes large the mean waiting time in the child job stealing system becomes *insensitive* with respect to the distribution of the number of child jobs. In addition, if $\mu_1 = \mu_2$ or μ_2 tends to infinity, we obtain for the mean waiting time $\rho/(\mu_1 - \lambda)$.

In the limit, the service time is the maximum of a set of exponentials and the expected service time is

$$E[J^{(c)}] = \sum_{k=0}^{m} p_k J_k, \tag{88}$$

where J_k, $k = 0, \ldots, m$, as the average time it takes for a parent job that will spawn k child jobs to be processed, with $J_0 = \frac{1}{\mu_1}$. For J_k, with $k \geq 1$, let $J^p \sim \exp(\mu_1)$, $J^1, \ldots, J^m \sim \exp(\mu_2)$ be independent random variables. We will provide a recursive formula for J_k using the following facts:

$$E[\min(J^p, J^1, \ldots, J^k)] = \frac{1}{\mu_1 + k\mu_2}, \tag{89}$$

$$P[J^p < \min(J^1, \ldots, J^k)] = \frac{\mu_1}{\mu_1 + k\mu_2}. \tag{90}$$

We now define recursively, for $k \geq 1$:

$$\begin{aligned} J_k &= \frac{1}{\mu_1 + k\mu_2} + \frac{\mu_1}{\mu_1 + k\mu_2} E[\max(J^1, \ldots, J^k)] + \frac{k\mu_2}{\mu_1 + k\mu_2} E[\max(J^p, J^1, \ldots, J^{k-1})], \\ &= \frac{1}{\mu_1 + k\mu_2} + \frac{\mu_1}{\mu_1 + k\mu_2} \frac{1}{\mu_2} \sum_{j=1}^{k} \frac{1}{j} + \frac{k\mu_2}{\mu_1 + k\mu_2} J_{k-1}. \end{aligned} \tag{91}$$

The limit of the mean response time is $E[T^{(c)}] = E[W^{(c)}] + E[J^{(c)}]$. The mean $E[J^{(c)}]$ does depend on the distribution $(p_1, \ldots, p_m)$ and the next proposition shows that this mean is maximized by the deterministic distribution. As $E[W^{(c)}]$ only depends on the mean $\tilde{p}$ of this distribution, this implies that the mean response time is maximized by the deterministic distribution as r tends to infinity, which is in strong contrast to the setting where r tends to zero (as the mean response time in an M/G/1 queue increases as job sizes become more variable).

Proposition 8.1. *If the mean number of child jobs equals $d \in \{1, 2 \ldots\}$, the service time is maximized by the deterministic distribution, that is,*

$$J_d \geq \sum_{n=0}^{m} p_n J_n, \tag{92}$$

such that $\sum_{n=1}^{m} n p_n = d$.

Proof. It suffices to argue that

$$J_n - J_{n-1} \geq J_{n+1} - J_n. \tag{93}$$

Due to (91) with $k = n + 1$, we have that $2J_n \geq J_{n+1} + J_{n-1}$ can be written as

$$(2\mu_1 + (n+1)\mu_2) J_n \geq 1 + \frac{\mu_1}{\mu_2} \sum_{j=1}^{n+1} \frac{1}{j} + (\mu_1 + (n+1)\mu_2) J_{n-1}. \tag{94}$$

By (91) with $k = n$, we get that this is equivalent to

$$\frac{2\mu_1 + (n+1)\mu_2}{\mu_1 + n\mu_2} \left(1 + \frac{\mu_1}{\mu_2} \sum_{j=1}^{n} \frac{1}{j} + n\mu_2 J_{n-1} \right) \geq 1 + \frac{\mu_1}{\mu_2} \sum_{j=1}^{n+1} \frac{1}{j} + (\mu_1 + (n+1)\mu_2) J_{n-1}$$

that is,

$$\frac{\mu_1 + \mu_2}{\mu_1 + n\mu_2} \left(1 + \frac{\mu_1}{\mu_2} \sum_{j=1}^{n} \frac{1}{j} + n\mu_2 J_{n-1} \right) \geq \frac{1}{n+1} \frac{\mu_1}{\mu_2} + (\mu_1 + \mu_2) J_{n-1}. \tag{95}$$

By multiplying with $\frac{\mu_1 + n\mu_2}{\mu_1 + \mu_2}$, we get

$$1 + \frac{\mu_1}{\mu_2} \sum_{j=1}^{n} \frac{1}{j} + n\mu_2 J_{n-1} \geq \frac{1}{n+1} \frac{\mu_1 + n\mu_2}{\mu_1 + \mu_2} \frac{\mu_1}{\mu_2} + (\mu_1 + n\mu_2) J_{n-1}. \tag{96}$$

Which, after dividing by μ_1, is equivalent to

$$\frac{1}{\mu_1} + \frac{1}{\mu_2}\left(\sum_{j=1}^{n}\frac{1}{j} - \frac{1}{n+1}\right) \geq \frac{n-1}{n+1}\frac{1}{\mu_1+\mu_2} + J_{n-1}. \tag{97}$$

By using the definition of J_{n-1} and multiplying by $\mu_1 + (n-1)\mu_2$, we get

$$\begin{aligned} &1 + (n-1)\frac{\mu_2}{\mu_1} + (n-1)\left(\sum_{j=1}^{n}\frac{1}{j} - \frac{1}{n+1}\right) + \frac{\mu_1}{\mu_2}\left(\sum_{j=1}^{n}\frac{1}{j} - \frac{1}{n+1}\right) \\ &\geq \frac{n-1}{n+1} + \frac{n-1}{n+1}\frac{(n-2)\mu_2}{\mu_1+\mu_2} + 1 + \frac{\mu_1}{\mu_2}\sum_{j=1}^{n-1}\frac{1}{j} + (n-1)\mu_2 J_{n-2}, \end{aligned} \tag{98}$$

which after simplification and division by $(n-1)\mu_2$ is equivalent to

$$\frac{1}{\mu_1} + \frac{1}{\mu_2}\left(\sum_{j=1}^{n}\frac{1}{j} - \frac{2}{n+1}\right) + \frac{1}{n-1}\frac{\mu_1}{\mu_2^2}\left(\frac{1}{n} - \frac{1}{n+1}\right) \geq \frac{n-2}{n+1}\frac{1}{\mu_1+\mu_2} + J_{n-2}. \tag{99}$$

Which holds if

$$\frac{1}{\mu_1} + \frac{1}{\mu_2}\left(\sum_{j=1}^{n}\frac{1}{j} - \frac{2}{n+1}\right) \geq \frac{n-2}{n+1}\frac{1}{\mu_1+\mu_2} + J_{n-2}. \tag{100}$$

Doing similar steps as between (97) and (99), we get that (100) is equivalent to

$$\frac{1}{\mu_1} + \frac{1}{\mu_2}\left(\sum_{j=1}^{n}\frac{1}{j} - \frac{3}{n+1}\right) + \frac{1}{n-2}\frac{\mu_1}{\mu_2^2}\left(\frac{1}{n-1} + \frac{1}{n} - \frac{2}{n+1}\right) \geq \frac{n-3}{n+1}\frac{1}{\mu_1+\mu_2} + J_{n-3}, \tag{101}$$

which is true if

$$\frac{1}{\mu_1} + \frac{1}{\mu_2}\left(\sum_{j=1}^{n}\frac{1}{j} - \frac{3}{n+1}\right) \geq \frac{n-3}{n+1}\frac{1}{\mu_1+\mu_2} + J_{n-3}, \tag{102}$$

and so on. In the end we get that $J_n - J_{n+1} \geq J_{n-1} - J_n$ holds if

$$\frac{1}{\mu_1} + \frac{1}{\mu_2}\left(\sum_{j=1}^{n}\frac{1}{j} - \frac{n}{n+1}\right) \geq J_0, \tag{103}$$

which is true as $J_0 = \frac{1}{\mu_1}$ and $\frac{1}{j} \geq \frac{1}{n+1}$. The result then follows immediately by concavity. □

Parent job stealing. Taking $r \to \infty$, the effect on the mean response and waiting times in the limit is shown in Figure 6. The expected waiting time $E[W^{(p)}]$ goes to zero and the mean response time $E[T^{(p)}]$ goes to the mean service time, $E[J^{(p)}]$, that is,

$$\lim_{r\to\infty} E[T^{(p)}(r)] = E[J^{(p)}], \tag{104}$$

defined by Equation (56).

9 MODEL VALIDATION

Mean field models are intended to capture the system behaviour as the number of servers in the systems tends to infinity. In this section we use simulation experiments to indicate that the system performance for a large finite system is very close to the fixed point of the mean field models. To prove that the sequence of the stationary measures of the finite systems weakly converge towards to the Dirac measure of the fixed point, one could leverage the methodology in [7]. In fact, if we truncate the queues to some large finite size B, proving the convergence of the sample paths over finite time scales towards the solution of the set of ODEs should be fairly straightforward using Kurtz's theorem [13], by defining a density dependent population process and showing that the drift is Lipschitz continuous. To show that the convergence can be extended to the stationary regime one also needs to establish global attraction of the fixed point. Global attraction is often proven using monotonicity arguments [5, 21, 24], but our multithreading models are not monotone (as the service time of a complete job does not necessarily have a decreasing hazard rate).

We consider different scenarios for varying probe rates and the two stealing strategies, for $N = 500$ with $\mu_1 = 1, \mu_2 = 2$ and child job distribution $\check{p} = [5, 4, 3, 2, 1]/15 \approx [0.33, 0.27, 0.20, 0.13, 0.07]$. Figure 1 shows the calculated waiting and response times for the fixed points of the mean field model (solid lines) and the simulations (dotted lines) for two different probe rates, $r = 1, 10$. The simulation started from an empty system and the system was simulated for $T = 10^5$ time units with a warm-up period of 33%. The 95% confidence intervals were computed based on 5 runs.

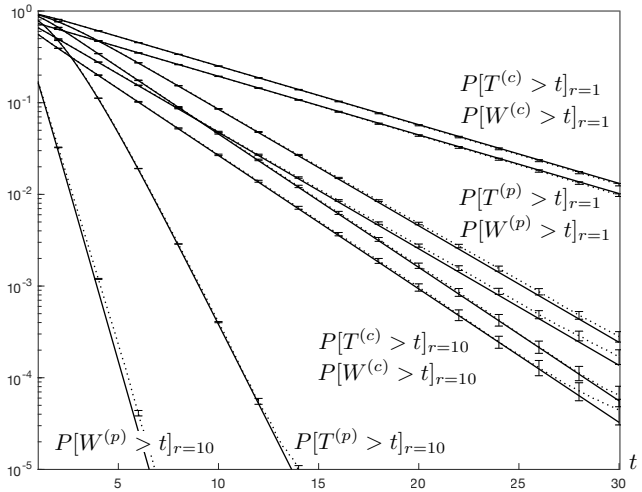

Fig. 1. Waiting and response times from the fixed points of the ODEs and the simulations.

We see that there is an excellent agreement between the ODE fixed point and simulation times for all settings, and that the ODE plots consistently lie close to the displayed confidence intervals. As expected, the response times for $r = 10$ are less than for $r = 1$, for both strategies.

Table 3 shows the mean field value and the relative errors obtained when comparing the mean response rate in a finite system with $N \in \{15, 30, 60, 125, 250, 500, 1000\}$ servers, under both stealing strategies with $r \in \{1, 10\}$, $\rho \in \{0.75, 0.85\}$ based on 20 runs. Overall the relative error tends to increase with ρ and r and decreases in N. In almost all of the scenarios considered, the mean field model is accurate to within 2% for $N \geq 125$. We see that for smaller values of $N \leq 30$ the error can be above 5%.

Table 3. Relative error of simulation results for $E[T^i(r)]$ for $i \in \{(c),(p)\}$, based on 20 runs

		$\rho = 0.75$		$\rho = 0.85$	
	N	sim. ± conf.	rel.err.%	sim. ± conf.	rel.err.%
$r = 1$					
(c)	15	4.6527 ± 5.62e-03	1.1571	7.5769 ± 1.92e-02	2.8210
	30	4.6512 ± 4.73e-03	1.1246	7.4344 ± 7.82e-03	0.8870
	60	4.6201 ± 3.72e-03	0.4483	7.4245 ± 1.09e-02	0.7527
	125	4.6033 ± 2.38e-03	0.0828	7.3902 ± 1.01e-02	0.2883
	250	4.6043 ± 9.84e-04	0.1037	7.3917 ± 3.16e-03	0.3080
	500	4.6035 ± 1.05e-03	0.0861	7.3659 ± 3.21e-03	0.0422
	1000	4.6002 ± 6.93e-04	0.0139	7.3712 ± 2.79e-03	0.0301
	∞	4.5995		7.3690	
(p)	15	3.4416 ± 3.22e-03	4.2954	4.9570 ± 1.04e-02	5.9664
	30	3.3620 ± 1.82e-03	1.8849	4.8390 ± 6.50e-03	3.4428
	60	3.3293 ± 1.63e-03	0.8935	4.7475 ± 3.38e-03	1.4865
	125	3.3195 ± 1.30e-03	0.5952	4.7035 ± 2.89e-03	0.5461
	250	3.3090 ± 8.72e-04	0.2765	4.6933 ± 1.97e-03	0.3291
	500	3.3045 ± 4.93e-04	0.1423	4.6865 ± 1.21e-03	0.1843
	1000	3.3027 ± 3.59e-04	0.0872	4.6830 ± 9.07e-04	0.1092
	∞	3.2998		4.6779	
$r = 10$					
(c)	15	2.9239 ± 1.90e-03	6.1114	4.1132 ± 7.19e-03	11.0528
	30	2.8372 ± 2.11e-03	2.9651	3.9128 ± 5.01e-03	5.6413
	60	2.7975 ± 1.35e-03	1.5253	3.8122 ± 2.49e-03	2.9265
	125	2.7729 ± 9.30e-04	0.6312	3.7490 ± 1.96e-03	1.2205
	250	2.7648 ± 6.89e-04	0.3400	3.7232 ± 1.29e-03	0.5225
	500	2.7587 ± 4.42e-04	0.1185	3.7209 ± 1.10e-03	0.4624
	1000	2.7573 ± 3.97e-04	0.0681	3.7085 ± 8.03e-04	0.1271
	∞	2.7555		3.7038	
(p)	15	2.1018 ± 1.12e-03	8.0698	2.5452 ± 2.31e-03	16.6288
	30	2.0165 ± 7.12e-04	3.6872	2.3586 ± 1.43e-03	8.0776
	60	1.9799 ± 4.00e-04	1.8054	2.2682 ± 8.22e-04	3.9376
	125	1.9601 ± 2.66e-04	0.7853	2.2223 ± 5.60e-04	1.8344
	250	1.9523 ± 1.79e-04	0.3835	2.2047 ± 4.07e-04	1.0259
	500	1.9493 ± 1.27e-04	0.2306	2.1931 ± 2.90e-04	0.4921
	1000	1.9466 ± 1.09e-04	0.0907	2.1877 ± 1.46e-04	0.2471
	∞	1.9448		2.1823	

10 NUMERICAL EXPERIMENTS

In this section we consider the performance, i.e., the mean response time of a job, for each stealing strategy followed by a comparison of the two.

10.1 Performance of child job stealing

Example A.1. For the mean waiting time with $r \to \infty$, we consider the effect of three child job distributions with mean 3, $\check{p}_1 \sim D(3)$, $\check{p}_2 \sim U(0,6)$ and $\check{p}_3$ defined by $p(m=1) = 5/7$ and $p(m=8) = 2/7$, with variances 0, 4, and 16, respectively. With $(\mu_1, \mu_2) = (1,2)$, we illustrate the mean waiting times in Figure 2 for $\rho = 0.75, 0.85$. For small probe rates, $r < 10^1$, we observe the increased variability of the child job distribution increases the waiting time, and in the limit see the system become insensitive to this distribution, as per Equation (87) for $\rho = 0.75, 0.85$ we have $E[W^{(c)}] = 0.75, 0.90$, respectively.

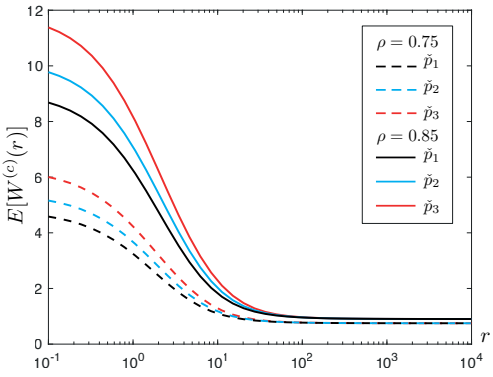

Fig. 2. Example A.1. $E[W^{(c)}(r)]$ with $r \to \infty$

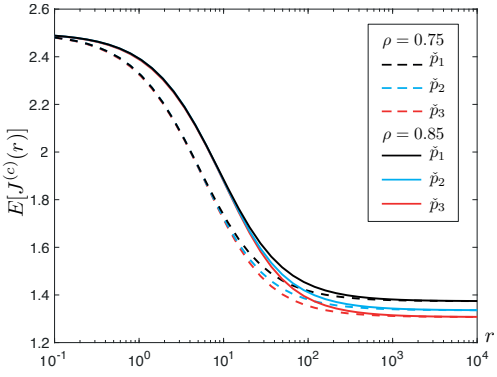

Fig. 3. Example A.1. $E[J^{(c)}(r)]$ with $r \to \infty$

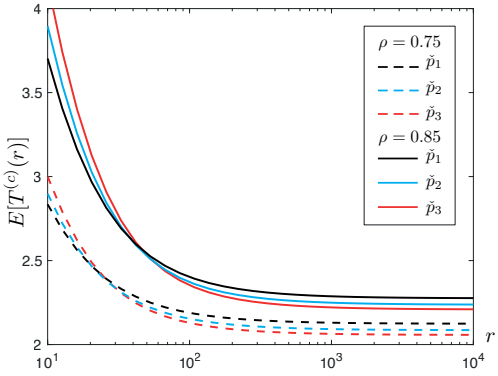

Fig. 4. Example A.1. $E[T^{(c)}(r)]$ with $r \to \infty$

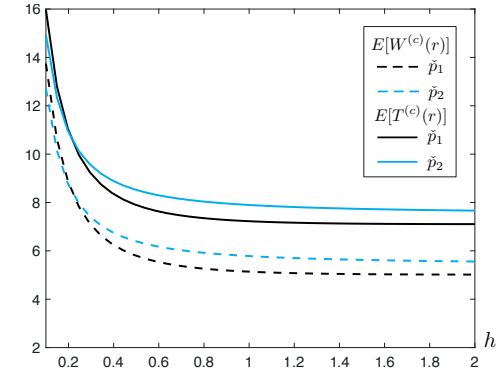

Fig. 5. Example A.2. Service ratio $h = \mu_1/\mu_2$

In Figure 3 we present the mean service time with $r \to \infty$. As expected for r close to zero the mean service time is 2.5, then for small $r < 10^1$, we observe insensitivity to $\check{p}$, whereas in the limit, we have for $\check{p}_1 : E[J^{(c)}] = 1.3742$, $\check{p}_2 : E[J^{(c)}] = 1.3360$, $\check{p}_3 : E[J^{(c)}] = 1.3076$, as per Equation (88). Thus in the limit, the child job distributions with positive variance $\check{p}_2$ and $\check{p}_3$ perform 2.78% and 4.85% better, respectively, than the deterministic child job distribution $\check{p}_1$, due to the result in Equation (92)

Combining the results in Figures 2 and 3, we obtain the mean response times illustrated in Figure 4 (noting the change in scale), and see that setting a probe rate $r \sim 10^2$ would provide performance

of $r = \infty$. We also observe the effect of the variability of $\check{p}$ changes from reducing the performance for small r, to improving to the performance for large r.

Example A.2. We consider the effect on performance of changing the parent to child job service ratio $h = \mu_1/\mu_2$ under a fixed load ρ and arrival rate λ. Given h, we have $\mu_2 = \frac{\lambda}{\rho}(\frac{1}{h} + \sum_{n=1}^{m} np_n)$ and $\mu_1 = \frac{\lambda}{\rho}(1 + h\sum_{n=1}^{m} np_n)$. As h is increased, the mean job size remains constant, the mean parent job size decreases and the mean child job size increases, and $\lim_{h\to\infty} 1/\mu_1 = 0$, $\lim_{h\to\infty} 1/\mu_2 = \rho(\lambda \sum_{n=1}^{m} np_n)^{-1}$. Figure 5 illustrates the mean waiting and response times for (λ, ρ, r)= (0.4, 0.95, 10) for $\check{p}_1 \sim U(2, 4)$ and $\check{p}_2 \sim U(4, 6)$. An improvement in performance, due to the reduction in waiting time, can be seen for h increasing to 1. For $h > 1$, the mean child job size has approached its limit and no further performance improvements are obtained. As expected, when the mean number of child jobs is decreased, from $\check{p}_2$ to $\check{p}_1$, i.e., the mean size of a child job is decreased, we see the performance improve. Not seen in the figure, the mean service times are equal for $\check{p}_1$ and $\check{p}_2$ and constant across h except for a slight increase when $h < 0.3$.

10.2 Performance of parent job stealing

Example B.1. For the distributions $\check{p}_1, \check{p}_2, \check{p}_3$ defined in Example A.1., with $(\mu_1, \mu_2) = (1, 2)$ and $\rho = 0.85$, in Figure 6 we observe that for $r \to \infty$ the waiting time is tends to zero and the mean response time approaches $E[T^{(p)}] = 2.5$, as per Equation (104). The impact on the mean response time reduces with an increased probe rate while as expected, the service time remains constant. The largest impact of variability on the waiting time occurs when $r < 1$, the child job distributions with positive variance $\check{p}_2$ and $\check{p}_3$ cause the mean waiting time to increase significantly compared to the deterministic distribution $\check{p}_1$. As r increases this effect is reduced, and no longer has an impact for $r > 10$.

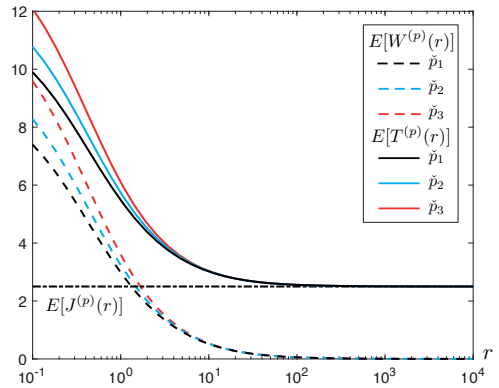

Fig. 6. Example B.1. Probe rate $r \to \infty$

Example B.2. Under parent job stealing, changing the proportion of the workload between parent and child jobs $h = \mu_1/\mu_2$ does not impact the performance of the system, as the job, the parent and any child jobs, is stolen together following a successful probe.

10.3 Child versus parent job stealing

Example C.1. We compare the two stealing strategies by illustrating the proportional difference in the mean response times for child job stealing compared to parent job stealing in Figure 7 under varying loads ρ and probe rates r for $\check{p} \sim D(2), D(4), D(6), D(8)$ and $(\mu_1, \mu_2) = (1, 2)$. Under high loads with small probe rates, parent job stealing shows the largest benefit in performance (blue region), due to the significant increase in mean waiting time under child job stealing. Increasing the number of child jobs in the system increases the orange region in which child stealing performs best and the white region shows where two strategies perform within ±6.7% of each other. When $\check{p} \sim D(8)$, we see that for $\rho = 0.5$ and $r = 20$, child job stealing performs approximately 50% better than parent job stealing, whereas for $\rho = 0.95$ and $r = 1$ parent job stealing performs approximately 100% better than child job stealing. We note that we obtain similar insights for other child job distributions.

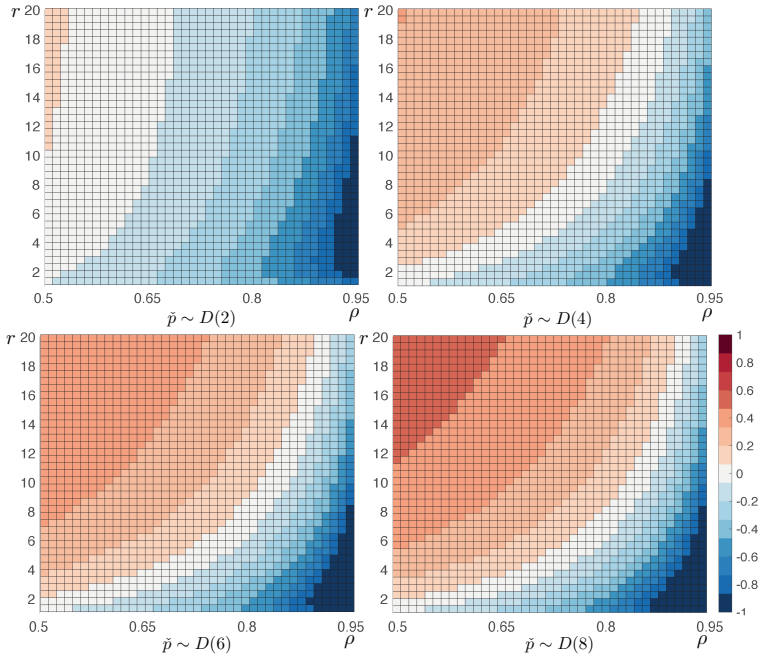

Fig. 7. Example C.1. Proportional difference in mean response time of (c) compared to (p): $(E[T^{(p)}] - E[T^{(c)}])/E[T^{(p)}]$

11 CONCLUSIONS AND MODEL GENERALIZATION

We introduced two mean field models for randomized work stealing in multithreaded computations in large systems, where parent jobs spawn child jobs. We proved the existence of a unique fixed point and showed that this fixed point can be computed easily using matrix analytic methods (by solving a single quasi-birth-death Markov chain). The accuracy of these models was illustrated using simulation experiments.

The two models correspond to two stealing strategies: one that involves the transfer of child jobs across servers; the other where parent jobs are transferred together with the child jobs that they spawn. Having derived expressions for the response time distributions for each strategy, we

investigated the impact of the probe rates, load, and child job size variability on performance with respect to the individual stealing strategies. We also studied the effect of changing the ratio of parent to child service rates and identified scenarios (low probe rate, high load) where parent stealing significantly outperformed child stealing, and scenarios (high probe rate, low load) where child stealing achieved a lower mean response time.

The models presented in this paper can be generalized in a number of manners. A first option is to allow for phase-type distributed service times for parent jobs and individual child jobs. This relaxation is easy for the parent stealing model (as the sum of several phase-type distributions is still a phase-type distribution [14]), but more involved for the child job stealing system. For instance, the steal rate $\lambda_c(r)$ for which we had an explicit formula in (30) now needs to be computed numerically. Nevertheless it seems likely that it still suffices to study the steady state of a single QBD Markov chain to assess the performance in a large-scale system.

A second relaxation is to allow for a finite number of server types, as opposed to having homogeneous servers. This generalization seems more challenging as the probability that a server is empty now depends on its type (instead of simply being $1-\rho$), which implies that an iterative approach may be needed to find the fixed point of the mean field model (where a QBD-type Markov chain is solved during each iteration).

Another generalization exists in allowing that multiple child jobs are transferred after a successful probe. Initial work in this direction indicates that the approach taken in this paper is still feasible in such case. A further relaxation would be to consider multigenerational multithreading, that is where child jobs can generate their own offspring jobs. In such case, using a QBD-type Markov chain seems problematic due to the required block size.

REFERENCES

[1] D.A. Bini, B. Meini, S. Steffe, and B. Van Houdt. 2006. Structured Markov chains solver: software tools. In *Proceeding from the 2006 workshop on Tools for solving structured Markov chains*. 14–es.

[2] M. Bladt and B.F. Nielsen. 2017. *Matrix-exponential distributions in applied probability*. Vol. 81. Springer.

[3] R. Blumofe, C. Joerg, B. Kuszmaul, C. Leiserson, K. Randall, and Y. Zhou. 1996. Cilk: An efficient multithreaded runtime system. *Journal of parallel and distributed computing* 37, 1 (1996), 55–69.

[4] R. Blumofe and C. Leiserson. 1999. Scheduling multithreaded computations by work stealing. *Journal of the ACM (JACM)* 46, 5 (1999), 720–748.

[5] M. Bramson, Y. Lu, and B. Prabhakar. 2012. Asymptotic independence of queues under randomized load balancing. *Queueing Syst.* 71, 3 (2012), 247–292.

[6] D. Eager, E. Lazowska, and J. Zahorjan. 1986. A comparison of receiver-initiated and sender-initiated adaptive load sharing. *Performance Evaluation* 6, 1 (1986), 53–68.

[7] N. Gast. 2017. Expected values estimated via mean-field approximation are $1/n$-accurate. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 1 (2017), 1–26.

[8] N. Gast and B. Gaujal. 2010. A mean field model of work stealing in large-scale systems. *ACM SIGMETRICS Performance Evaluation Review* 38, 1 (2010), 13–24.

[9] T. Gautier, X. Besseron, and L. Pigeon. 2007. Kaapi: A thread scheduling runtime system for data flow computations on cluster of multi-processors. In *Proceedings of the 2007 International workshop on Parallel symbolic computation*. 15–23.

[10] G. Golub and C. Van Loan. 2012. *Matrix computations*. Vol. 3. JHU press.

[11] M. Harchol-Balter. 2021. Open problems in queueing theory inspired by datacenter computing. *Queueing Systems* 97, 1 (2021), 3–37.

[12] G. Horváth, B. Van Houdt, and M. Telek. 2014. Commuting matrices in the queue length and sojourn time analysis of MAP/MAP/1 queues. *Stochastic Models* 30, 4 (2014), 554–575.

[13] T.G. Kurtz. 1981. *Approximation of population processes*. Vol. 36. SIAM.

[14] G. Latouche and V. Ramaswami. 1999. *Introduction to matrix analytic methods in stochastic modeling*. Vol. 5. SIAM.

[15] W. Minnebo, T. Hellemans, and B. Van Houdt. 2017. On a class of push and pull strategies with single migrations and limited probe rate. *Performance Evaluation* 113 (2017), 42–67.

[16] W. Minnebo and B. Van Houdt. 2014. A fair comparison of pull and push strategies in large distributed networks. *IEEE/ACM Transactions on Networking (TON)* 22, 3 (2014), 996–1006.

[17] R. Mirchandaney, D. Towsley, and J.A. Stankovic. 1990. Adaptive load sharing in heterogeneous distributed systems. *Journal of parallel and distributed computing* 9, 4 (1990), 331–346.
[18] M.F. Neuts. 1981. *Matrix-geometric solutions in stochastic models: an algorithmic approach.* John Hopkins University Press.
[19] T. Ozawa. 2006. Sojourn time distributions in the queue defined by a general QBD process. *Queueing Systems* 53, 4 (2006), 203–211.
[20] A. Robison, M. Voss, and A. Kukanov. 2008. Optimization via reflection on work stealing in TBB. In *2008 IEEE International Symposium on Parallel and Distributed Processing*. IEEE, 1–8.
[21] S. Shneer and A. Stolyar. 2021. Large-scale parallel server system with multi-component jobs. *Queueing Systems* 98, 1 (2021), 21–48.
[22] I. Van Spilbeeck and B. Van Houdt. 2015. Performance of rate-based pull and push strategies in heterogeneous networks. *Performance Evaluation* 91 (2015), 2–15.
[23] M.S. Squillante and R.D. Nelson. 1991. Analysis of task migration in shared-memory multiprocessor scheduling. *SIGMETRICS Perform. Eval. Rev.* 19, 1 (1991), 143–155.
[24] B. Van Houdt. 2019. Global attraction of ODE-based mean field models with hyperexponential job sizes. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3, 2 (2019), 1–23.
[25] B. Van Houdt. 2019. Randomized work stealing versus sharing in large-scale systems with non-exponential job sizes. *IEEE/ACM Transactions on Networking* 27 (2019), 2137–2149. Issue 5.
[26] N. Wirth. 1996. Tasks versus threads: an alternative multiprocessing paradigm. *Software - Concepts and Tools* 17 (01 1996), 6–12.
[27] J. Yang and Q. He. 2018. Scheduling parallel computations by work stealing: A survey. *International Journal of Parallel Programming* 46, 2 (2018), 173–197.