Performance Analysis of Workload Dependent Load Balancing Policies

TIM HELLEMANS, TEJAS BODAS, BENNY VAN HOUDT, University of Antwerp, Belgium

Load balancing plays a crucial role in achieving low latency in large distributed systems. Recent load balancing strategies often rely on replication or use placeholders to further improve latency. However assessing the performance and stability of these strategies is challenging and is therefore often simulation based. In this paper we introduce a unified approach to analyze the performance and stability of a broad class of workload dependent load balancing strategies. This class includes many replication policies, such as replicate below threshold, delayed replication and replicate only small jobs, as well as strategies for fork-join systems.

We consider systems with general job size distributions where jobs may experience server slowdown. We show that the equilibrium workload distribution of the cavity process satisfies a functional differential equation and conjecture that the cavity process captures the limiting behavior of the system as its size tends to infinity.

We study this functional differential equation in more detail for a variety of load balancing policies and propose a numerical method to solve it. The numerical method relies on a fixed point iteration or a simple Euler iteration depending on the type of functional differential equation involved. We further show that additional simplifications can be made if certain distributions are assumed to be phase-type.

Various numerical examples are included that validate the numerical method and illustrate its strength and flexibility.

Additional Key Words and Phrases: Workload; Redundancy; Large Scale Computer Network; Differential Equation; Fixed Point Equation; Load Balancing

ACM Reference Format:
Tim Hellemans, Tejas Bodas, Benny Van Houdt. 2019. Performance Analysis of Workload Dependent Load Balancing Policies. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 2, Article 35 (June 2019), 35 pages. https://doi.org/10.1145/3326150

1 INTRODUCTION

Latency minimization is an important consideration in large scale data networks, server farms, cloud and grid computing, etc. A key role in achieving low latency is played by the load balancer responsible for distributing the jobs among the available servers. Popular load balancing schemes include the *join-shortest-queue among d randomly selected queues* (JSQ(d)) [1, 5, 16, 22] and the *join-idle-queue* (JIQ) [7, 15, 21] scheme. Under these schemes any incoming job is immediately assigned to a single server in the system.

A recent trend to further reduce latency is to use redundancy, that is, to assign an incoming job to multiple servers by distributing replicas of a job among the servers [2]. Initially this form of redundancy was introduced to combat unexpected server slowdowns, that is, a short job may suddenly experience an exceptionally long delay even if the server has low load. When redundancy

Author's address: Tim Hellemans, Tejas Bodas, Benny Van Houdt, University of Antwerp, Middelheimlaan 1, Antwerp, B-2020, Belgium, tim.hellemans@uantwerpen.be,tejaspbodas@gmail.com,benny.vanhoudt@uantwerpen.be.

2476-1249/2019/6-ART35 $15.00
https://doi.org/10.1145/3326150

is used, one can either cancel all remaining replicas as soon as one completes service [9] or as soon as a replica starts service [4, 11]. The latter is useful to reduce the time that a job spends waiting in the queue, but is less effective when servers are subject to unexpected slowdowns. Fork-join based systems are another area where redundancy has been introduced to reduce latency [12, 13, 19]. In a fork-join system, a task is subdivided into sub-tasks which are executed on different servers and finally merged back as soon as the sub-tasks have been completed. Thus if one sub-task is delayed, so is the complete task. By introducing redundancy it suffices that only a subset of the sub-tasks complete.

To assess the performance of these load balancing schemes most prior work relied on mean-field models, that is, studied the limiting behavior as the number of servers in the system becomes large under the assumption of asymptotic independence (an assumption that is very hard to prove for general service times, see [6]). In case of JSQ(d) and JIQ, where jobs are assigned immediately to a single server, the stability condition is simple and the system state is fully captured by the queue length at the different servers (plus the remaining job size in case of general job sizes). For systems with redundancy such a state description no longer works and even the system stability becomes complicated as replicas increase the actual workload and too much replication can easily lead to system instability [19].

Most prior analytical work on systems with redundancy focused on the redundancy-d (Red(d)) policy which replicates incoming jobs on d randomly selected servers, where the remaining replicas are either cancelled as the first replica starts or completes service. Product forms for the system state of LL(d) resp. Red(d) under the assumption of exponential job sizes resp. exponential job sizes and replicas that have independent sizes were presented in [4, 9]. Furthermore, in [3] a recent token based framework to analyse product forms and relevant performance measures for a variety of central queue based multi server models including LL(d) and Red(d) models was also introduced. A mean-field model for Red(d) with cancellation on completion was developed in [9] for independent replicas and in [10] for identical replicas. Red(d) with cancellation on start, which corresponds to assigning the job to the least loaded server, was analysed in [11]. The stability issue of Red(d) with cancellation on completion was avoided in [8] by the RIQ policy, which replicates incoming jobs only on the idle servers among a set of d randomly selected servers (to mitigate the effect of server slowdown). This also simplified the performance analysis somewhat as existing results on vacation queues could be leveraged.

Another important contribution of [8] exists in introducing the S&X model. Under this model any replica has the same inherent job size X, but the actual service time of a replica on a server equals S times X, where S represents the slowdown that is assumed independent among replicas (as it depends on the server). This model is clearly closer to reality than assuming that all the replicas have independent job sizes (which is known to yield misleading insights such as more replication always reduces response times).

While [11] and [10] studied two different systems with redundancy, both develop a mean-field model that studies the evolution of the workload at a server. In this paper we show that a very broad class of load balancing policies that rely on the workload information at a set of randomly selected servers can be analysed in a unified manner. More specifically, using the cavity process introduced in [5] we show that the workload distribution at a server is the solution of a functional differential equation (FDE) under the assumption of asymptotic independence. We further study this FDE for a variety of load balancing policies belonging to this class under the S&X model. These include many load balancing schemes of practical interest for which no analytical method to assess their performance existed so far. Examples include policies that use delayed replication, replicate only on servers with a workload below some threshold, replicate only small jobs, replication in fork-join queues, etc.

The paper makes the following contributions:

(1) We define the cavity process for a broad class of workload dependent load balancing policies, characterise its transient evolution and show that its equilibrium environment is the solution of an FDE.
(2) We show that many practical load balancing policies fit within our class of workload dependent policies and study their FDEs under the S&X model with general job size and slowdown distributions.
(3) We propose different numerical methods to solve these FDEs, present numerical results for both the stability and response times and validate the accuracy of our approach using simulation.
(4) We demonstrate that the numerical method can be further simplified if some of the distributions are phase-type (PH).

With respect to the numerical method, we distinguish four different types of FDEs:

- *Type 1:* Future independent policies with unknown system load.
- *Type 2:* Future dependent policies with unknown system load.
- *Type 3:* Future independent policies with known system load.
- *Type 4:* Future dependent policies with known system load.

For each policy we obtain an FDE of the form $\bar{F}'(w) = T(\bar{F}(u), u \in A_w)$. For the future independent policies we have $A_w \subseteq [0, w]$ (Type 1, 3), which allows us to solve these policies using a simple forward Euler scheme. For the future dependent policies $A_w \nsubseteq [0, w]$ (Type 2,4), for these policies we rely on a fixed point iteration to obtain the equilibrium workload distribution. The second distinction is made on whether or not the system load, $\bar{F}(0)$ is known (Type 3,4) or unknown (Type 1,2). When the load is unknown we use $\bar{F}(\infty) = 0$ as a boundary condition, otherwise we simply use the boundary condition on $\bar{F}(0)$. All code used to generate the figures used in the numerical experiments can be found at https://github.com/THellemans/workload_dependent_policies.

The paper is organized as follows: in Section 2 we describe the model considered in this paper in more detail. The terminology of the queue at the cavity is introduced in Section 3, we then define some common notation in Section 4. This is followed by the analysis of the transient and equilibrium behaviour of the queue at the cavity in Section 5. We then apply our general result to many examples in Section 6. The equations for these examples are further studied when certain random variables are PH distributed in Section 7. In Section 8 we propose a numerical method to find the equilibrium distribution and the stability region from the results of Section 6 and 7. Results that validate our approach are given in the Appendix, where we also elaborate on the redundancy based policies in case there is no slowdown.

2 MODEL DESCRIPTION

We consider a generic power-of-d system consisting of N identical, infinite buffer servers which serve jobs in a FCFS manner (here N is usually assumed to be large). Arrivals occur according to a Poisson(λN) process and the service rate at each server equals one. Whenever a job arrives, d distinct servers are chosen uniformly at random (with or without replacement). The job then creates some (or possibly no) added work on each of the d chosen servers depending on the load balancing policy used. The policy is chosen such that the added work (i.e. the actual arrival size) solely depends on the workload at each of the chosen servers (and other variables, independent of the chosen servers). For the load balancing policies considered in this paper, this added work consists of either the arriving job, partial execution of the job or other overheads due to placeholders as in the $LL(d, k, \delta)$ policy studied in Section 6.5. We shall henceforth refer to this type of model as

a workload dependent load balancing policy. Note that for this model with finite N, the process which only keeps track of the workload at each server, is a Markov process.

3 CAVITY PROCESS

We employ the cavity process methodology introduced in [5] to formulate a general method to obtain the transient and equilibrium workload distribution for a workload dependent load balancing policy in the mean-field regime. We first provide some intuition as to why the study of a queue at the cavity might be of interest. Looking at the many server system, instead of attempting to capture the global evolution of all N workload distributions, we single out one queue which we will henceforth refer to as the queue at the cavity. It is not hard to see that, as arrivals occur at rate λN and each arrival selects d queues, the queue at the cavity is selected with a rate equal to λd. Every time it is selected, we have to add some (or possibly no) work to it where the amount of work depends on the workload of the d selected queues. As we are not keeping track of the workload at any of the $d-1$ other selected queues, we simply generate their workload as a random variable which is independent of but identically distributed as the workload of the queue at the cavity at the time of the arrival. This method is known to yield exact results as $N \to \infty$ for some policies (those for which Conjecture 3.5 holds) and can often be used as a good approximation for sufficiently high values of N (see Appendix A.1)

In the cavity process method, potential arrivals occur according to a Poisson(λd) process. Whenever a potential arrival occurs, we create $d-1$ random variables with the same distribution as the queue at the cavity, add the actual arrival size to the queue at the cavity and discard these $d-1$ random variables again. Concretely: let $U_1, \ldots, U_d$ denote the (i.i.d.) workloads at the d chosen servers just before the potential arrival, where w.l.o.g. U_1 represents the queue at the cavity. Suppose we are given some additional random variables $V_1, \ldots, V_r$ (e.g., job size or server slowdown variables) that influence the added work. Then, we denote by $\boldsymbol{Q}(U_1, \ldots, U_d, V_1, \ldots, V_r)$ the random variable which represents the new workload in the queue at the cavity U_1. We call a potential arrival to U_1 an actual arrival if and only if $\boldsymbol{Q}(U_1, \ldots, U_d, V_1, \ldots, V_r) > U_1$. Note that while potential arrivals occur according to a Poisson(λd) process, the rate of actual arrivals strongly depends on the chosen policy and is generally hard to compute. Furthermore, depending on the load balancing policy, the actual arrival comprises of jobs that are either served completely at this server, jobs that are partially executed at the server or even other overhead like fetching a job which is no longer available. To illustrate what $\boldsymbol{Q}$ signifies, we present a few simple examples for policies which have been studied before.

Example 3.1. Consider the LL(d) policy studied in [11], where an incoming job of a certain size joins the least loaded server among d selected servers. In this case $r = 1$, $V_1 = X$ is the job size and $\boldsymbol{Q}(U_1, \ldots, U_d, X)$ is equal to $U_1 + X$ if $U_1 < \min_{i=2}^{d} U_i$ and it is equal to X with probability $\frac{1}{m}$ if $U_j = 0$ for exactly m choices of j including $j = 1$. Otherwise $\boldsymbol{Q}(U_1, \ldots, U_d, X) = U_1$.

Example 3.2. Two other examples are Red(d) with independent resp. identical replicas as studied in [9] resp. [10], where an incoming job replicates itself onto d servers and experiences an independent resp. identical service time on each server. The job is then cancelled as soon as one of the replicas finishes. For the case of independent replicas, we have $r = d$ and $V_i = X_i$ where $X_i, i = 1, \ldots, d$ are the i.i.d. job size variables. In this case, we have $\boldsymbol{Q}(U_1, \ldots, U_d, X_1, \ldots, X_d) = \max\{U_1, \min_{i=1}^{d}\{U_i + X_i\}\}$, indeed, a replica of the job finishes service by time $\min_{i=1}^{d}\{U_i + X_i\}$. For the case when the replicas are identical, we have $r = 1$ and $V_1 = X$ where X is the job size. A replica finishes service by time $\min_{i=1}^{d}\{U_i\} + X$, which yields $\boldsymbol{Q}(U_1, \ldots, U_d, X) = \max\{U_1, \min_{i=1}^{d}\{U_i\} + X\}$.

Definition 3.3 (Cavity Process). Let $\mathcal{H}(t)$, $t \geq 0$, be a set of probability measures on $[0, \infty)$ called the *environment process*. The *cavity process* $U^{\mathcal{H}(\cdot)}(t)$, $t \geq 0$, takes values in $[0, \infty)$ and is defined as follows. Potential arrivals occur according to a Poisson process with rate λd. When a potential arrival occurs at time t, the cavity process $U^{\mathcal{H}(\cdot)}(t)$ becomes $Q(U^{\mathcal{H}(\cdot)}(t-), U_2, \ldots, U_d, V_1, \ldots, V_r)$. Here $U_2, \ldots, U_d$ are $d-1$ independent random variables with law $\mathcal{H}(t-)$, and $V_1, \ldots, V_r$ are random variables which are independent of the process $U^{\mathcal{H}(\cdot)}(\cdot)$. The cavity process decreases at rate one during periods without arrivals and is lower bounded by zero.

We now define the cavity process associated to the equilibrium environment process, which is such that the cavity process itself has distribution $\mathcal{H}(t)$ at time t:

Definition 3.4 (Equilibrium Environment). When a cavity process $U^{\mathcal{H}(\cdot)}(\cdot)$ has distribution $\mathcal{H}(t)$ for all $t \geq 0$, we say that $\mathcal{H}(\cdot)$ is an *equilibrium environment process*. Further, a probability measure $\mathcal{H}$ is called an *equilibrium environment* if $\mathcal{H}(t) = \mathcal{H}$ for all t and $U^{\mathcal{H}(\cdot)}(t)$ has distribution $\mathcal{H}$ for all t.

A modularized program for analyzing load balancing systems by using the cavity process method was presented in [5]. In this program, one essentially needs to show asymptotic independence, which allows to assume that the workloads at the different queues become independent random variables and justifies that the behaviour of the entire system can be described by the behaviour of the queue at the cavity. One then needs to find a defining equation for the equilibrium behaviour of the queue at the cavity. This equation is given by (3) for our model. We use this equation to study several workload dependent load balancing policies. As will become apparent further on, all workload dependent load balancing policies which have been studied in the mean-field regime thus far can be analysed using this approach.

The asymptotic independence between the different queues is something which is very difficult to prove in general. Known proof techniques only exist for the LL(d) policy, the JSQ(d) policy under decreasing hazard rate (DHR) service requirements and the fork-join system. We believe that for the policies under consideration, the queues in the limiting regime satisfy this asymptotic independence property and then proceed with applying the modularized program. The remarkable accuracy between the performance measures obtained using our method and those obtained via simulation (see Appendix A.1) supports our belief that the following conjecture holds:

Conjecture 3.5. *Consider a workload dependent load balancing policy with N servers (each server has an FCFS discipline) as considered in Section 6 and assume this system is uniformly stable for sufficiently large N. Then, in the large N limit, the system has a unique equilibrium workload distribution under which any finite number of queues are independent. Moreover this equilibrium distribution is given by the equilibrium distribution of the associated cavity process.*

Remark. *The results in this paper characterize the queue at the cavity associated to workload dependent policies. In case Conjecture 3.5 fails to hold for a policy, one can still analyse the associated queue at the cavity regardless and this may be used as an (accurate) approximation for the actual model.*

4 NOTATION

For a random variable Y, we denote its cumulative distribution function (cdf) and complementary cdf (ccdf) by F_Y and $\bar{F}_Y$. Throughout, we assume all random variables Y used have no singular part and can therefore be decomposed into a continuous Y_c and a discrete part Y_d. Y_c has a pdf f_{Y_c} and Y_d can take values y_n with probability p_n where $p_n = \mathbb{P}\{Y_d = y_n\}$ with $\int_0^\infty f_{Y_c}(u)\,du + \sum_n p_n = 1$. In this case, for any function $h : [0, \infty) \to \mathbb{R}$ we have $\int_0^\infty h(u) dF_Y(u) = \int_0^\infty h(u) f_{Y_c}(u)\,du + \sum_n h(y_n) p_n$.

For ease of notation we write $Q(Y)$ instead of $Q(Y, U_2, \ldots, U_d, V_1, \ldots, V_r)$ whenever the random variables $U_2, \ldots, U_d$ and $V_1, \ldots, V_r$ are clear from the context. In words, given a workload of Y at the cavity queue just before the potential arrival, $Q(Y)$ indicates the *effective workload* in the cavity queue after the potential arrival. The *effective workload* at a server is the actual work that will be executed at the server and thus ignores jobs that were added to a queue and subsequently cancelled without receiving any service. In most cases we have as Y the workload at the queue at the cavity right before an arrival at time t: $Y = U^{\mathcal{H}(\cdot)}(t-)$ or the equilibrium workload distribution of the queue at the cavity: $Y = U^{\mathcal{H}}$. Furthermore both $U^{\mathcal{H}(\cdot)}(t-)$ and $U^{\mathcal{H}}$ will often be replaced by U_1 or U.

We denote by $f(t,\cdot)$ the pdf for the workload of the queue at the cavity at time t, $F(t,\cdot)$ its cdf and $\bar{F}(t,\cdot)$ its ccdf. In equilibrium, we drop the time dependence and simply denote the pdf, cdf and ccdf by $f(\cdot)$, $F(\cdot)$ and $\bar{F}(\cdot)$. For any workload dependent load balancing policy, we denote by R the response time random variable for the queue at the cavity at equilibrium. This response time can be found by generating d i.i.d. random variables $U_1, \ldots, U_d$ with distribution F and compute the response time given these random variables as the workload at the d chosen queues. For example for the LL(d) policy, if we let X denote a random variable which is distributed as the job size, one finds that $R = \min_{i=1}^{d}\{U_i\} + X$ (many more examples can be found in Section 6).

5 MEAN-FIELD ANALYSIS

5.1 Transient Behaviour

We start with the transient behavior. Note that at each time t, the pdf of the workload of the queue at cavity, i.e., $f(t,\cdot)$ integrates to $\int_0^\infty f(t,u)\,du = \bar{F}(t,0)$. As typically, $\bar{F}(t,0) < 1$ we have a point mass at zero which is equal to $F(t,0)$. For the transient behaviour, we obtain the following Partial Delayed Integro Differential Equation (PDIDE):

Theorem 5.1. *The workload of the queue at the cavity satisfies the following PDIDE:*

$$\begin{aligned}\frac{\partial f(t,w)}{\partial t} - \frac{\partial f(t,w)}{\partial w} = -\lambda d\Bigg[& f(t,w)\mathbb{P}\{Q(U^{\mathcal{H}(\cdot)}(t-)) > w \mid U^{\mathcal{H}(\cdot)}(t-) = w\} \\ & - F(t,0)\mathbb{P}\{Q(U^{\mathcal{H}(\cdot)}(t-)) = w \mid U^{\mathcal{H}(\cdot)}(t-) = 0\} \\ & - \int_0^w f(t,u)\mathbb{P}\{Q(U^{\mathcal{H}(\cdot)}(t-)) = w \mid U^{\mathcal{H}(\cdot)}(t-) = u\}\,du\Bigg] \end{aligned} \tag{1}$$

$$\frac{\partial F(t,0)}{\partial t} = -\lambda d\Big[F(t,0)\mathbb{P}\{Q(U^{\mathcal{H}(\cdot)}(t-)) = w \mid U^{\mathcal{H}(\cdot)}(t-) = 0\} + f(t,0^+)\Big], \tag{2}$$

for $w > 0$, where $f(t,w^+) = \lim_{v \downarrow w} f(t,v)$.

Proof. The proof is similar to the proof of Theorem 3.4 in [11] and is presented in Appendix A.4. □

5.2 Equilibrium Environment

To compute the equilibrium distribution we need to take the limit $t \to \infty$, thereby leaving out the dependence on t. In particular, we have $\frac{\partial f(s)}{\partial t} = 0$. We now directly derive a Functional Differential Equation (FDE) for the workload distribution from Theorem 5.1.

Theorem 5.2. *The equilibrium workload distribution of the queue at the cavity satisfies the following FDE:*

$$\bar{F}'(w) = -\lambda d\mathbb{P}\left\{U^{\mathcal{H}} \le w, Q(U^{\mathcal{H}}) > w\right\}. \tag{3}$$

Proof. For convenience we write U for $U^{\mathcal{H}(\cdot)}$. From (1) we readily obtain the following by integrating w.r.t. w once:

$$f(0) - f(w) = -\lambda d \int_0^w \Bigg[\mathbb{P}\{Q(U) > u, U = u\} - \mathbb{P}\{Q(U) = u, U = 0\} - \int_0^u \mathbb{P}\{Q(U) = u, U = v\}\, dv \Bigg] du. \tag{4}$$

The equality in (2) reduces to the boundary condition:

$$f(0) = \lambda d \mathbb{P}\{Q(0) > 0\} F(0),$$

using the fact that $\bar{F}'(w) = -f(w)$ we obtain from (4):

$$\bar{F}'(w) = -\lambda d \Bigg[F(0)\mathbb{P}\{Q(0) > 0\} + \int_0^w \mathbb{P}\{Q(U) > u, U = u\}\, du \Bigg] + \lambda d \int_0^w \mathbb{P}\{Q(U) = u, U = 0\}\, du + \lambda d \int_0^w \int_0^u \mathbb{P}\{Q(U) = u, U = v\}\, dv\, du. \tag{5}$$

Note that the first line in (5) is the rate of all possible upward jumps of U after a potential arrival when the workload in the cavity queue just before the potential arrival satisfies $U \in [0, w]$. The first term in the second line in (5) is the rate at which U jumps to somewhere below w when $U = 0$ at the time of a potential arrival. The last term in the second line in (5) is the rate at which U jumps up to somewhere below w while $U \in (0, w]$. The last two events are subsets of the first event and it can be observed that the difference of these events is the rate at which the cavity process jumps to a workload larger than w for $U \in [0, w]$. This yields equality (3). □

Remark. *The left hand side of* (3) *is* $\bar{F}'(w) = -f(w)$*, where* $f(w)$ *is the down-crossing rate through* w *while the right hand side is minus the up-crossing rate through* w*.*

6 LOAD BALANCING POLICIES

While our main result (Theorem 5.2) is applicable for any workload dependent load balancing policy as described in Section 2, in this section we specialize this result for some practical workload dependent policies. In many classic load balancing methods (like e.g. LL(d) and SQ(d)), a job is only sent to one server and its processing time solely depends on the speed of that server. There are however many load balancing policies, in particular those which employ some type of redundancy, which use the processing power of multiple servers in order to complete service. In this case, the question arises as to how one should treat the processing time at the different servers. Two popular choices are to assume that the processing time at the chosen servers are independent (see e.g. [9]) or that the processing times are identical (see e.g.[10]). Recently the S&X model was introduced in [8], this model is a combination of identical and independent replicas, each job has a size X which is identical over all chosen servers and a slowdown S which is independent over the chosen servers. In this section we analyse all considered policies in a setting which is a generalization of the S&X model which we explain shortly. In this section, we also present various numerical results for these policies to outline some important features. Simulation experiments that validate our approach can be found in Appendix A.1 and A.2.

Each job has an inherent size $X > 0$ and on each of the servers a job replica experiences some arbitrary slowdown denoted by the variable S_i. Thus each arrival is defined by a random job size variable X and d i.i.d. slowdown random variables $S_1, \ldots, S_d$. Using the notation of Section 5.1 we set $r = d + 1$, $V_i = S_i$ and $V_{d+1} = X$. While the actual processing time of the i-th replica in the S&X model of [8] then equals $S_i X$, we consider a more general setting. We assume that there

exists some function $g : [0,\infty) \times (0,\infty) \to (0,\infty)$ which is non-decreasing in both components such that if an arrival occurs, it has size $g(S_i, X)$ on the i^{th} chosen server. For any $s, x > 0$, define $g_x(s) = g(s,x)$ and assume it is a strictly increasing, continuous function. Note that in particular its inverse exists and we assume the inverse is differentiable. In our numerical experiments we set $g(S,X) = X + SX$, where S and X are generally distributed random variables with X the inherent job size and S the slowdown variable, such that the processing time cannot be less than X irrespective of the slowdown.

Example 6.1. Consider the Red(d) policy where, at each arrival, the job is replicated on d servers. Suppose the workload resp. the slowdown at each of the d servers is given by $U_1, \ldots, U_d$ resp. $S_1, \ldots, S_d$ and the job size is X. In this case we find that the workload U_1 is increased to:

$$\boldsymbol{Q}(U_1) = \max\{U_1, \min_{i=1}^{d}\{U_i + g(S_i, X)\}.$$

Moreover, the response time is given by:

$$R = \min_{i=1}^{d}\{U_i + g(S_i, X)\}.$$

Before proceeding with the analysis of the different load balancing policies, we outline some more notations used throughout the paper. For any sequence of random variables $Y_1, \ldots, Y_n$, let $Y_{(k)}$ denote its k'th order statistic such that $Y_{(1)} \le Y_{(2)} \le \cdots \le Y_{(n)}$, and ties are broken at random (this is mainly used in the Appendix). In the $S\&X$ setting, we define $R_x = U + g_x(S)$ as the sojourn time of a job of size x if it is sent to a single server with workload U and slowdown S. In this case, its ccdf is given by:

$$\begin{aligned}\bar{F}_{R_x}(w) &= \bar{F}_{g_x(S)}(w) + \int_0^w \bar{F}(u) f_{g_x(S)}(w-u)\,du \\ &= \bar{F}_S(g_x^{-1}(w)) + \sum_{u \le w} \bar{F}(w-u)\mathbb{P}\{S_d = g_x^{-1}(u)\} + \\ &\int_0^w \bar{F}(w-u) f_{S_c}(g_x^{-1}(u)) \cdot (g_x^{-1})'(u) I_{g_x([0,\infty))}(u)\,du, \end{aligned} \tag{6}$$

where the second equality follows from the fact that the pdf of $g_x(S_c)$ is given by $f_{g_x(S_c)}(w) = f_{S_c}(g_x^{-1}(w)) \cdot (g_x^{-1})'(w) I_{g_x([0,\infty))}(w)$ (here $I_A(u)$ equals one if $u \in A$ and zero otherwise). Moreover we denote by $\tilde{X} = g(S,X)$ the job size distribution at a single server. Analogously to (6), we find for $R_{\tilde{X}} = U + \tilde{X}$:

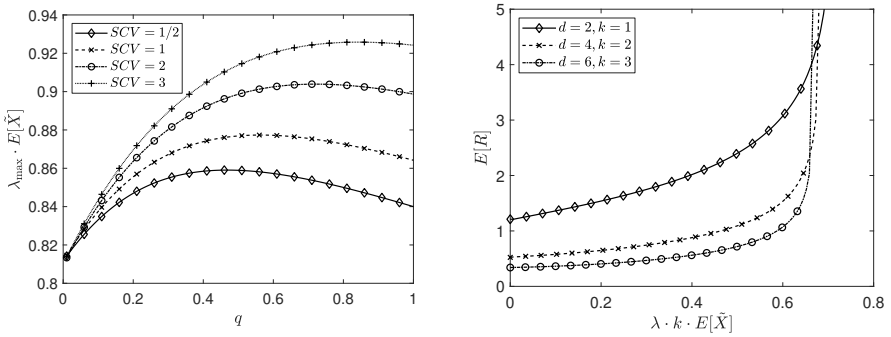
$$\bar{F}_{R_{\tilde{X}}}(w) = \bar{F}_{\tilde{X}}(w) + \int_0^w \bar{F}(u) f_{\tilde{X}}(w-u)\,du, \tag{7}$$

where the integral can again be split into a discrete and continuous part.

6.1 Type 1 : Red(d,k,δ)

In this section, we analyse the redundancy based policy Red(d,k,δ). Under this policy, an arriving job of size $k \cdot X$ selects d servers uniformly at random and places an identical replica of size X at each of the d servers. When any k of the d replicas have received service, the other redundant replicas are cancelled. Additionally we assume that the cancellation of redundant replicas requires a constant time $\delta \ge 0$. In other words, this means that once the k'th replica has been completed, the other servers continue working on remaining replicas (if they happen to be in service at that server) for a time δ. We indicate how this policy is used in practice in Appendix A.3.1.

We now show that the FDE in Theorem 5.2 reduces to a Delayed Integro Differential Equation (DIDE) without a boundary condition, meaning it is a Type 1 policy.



(a) $\lambda_{\max}\mathbb{E}[\tilde{X}]$ versus q for different slowdown distributions, $d = 2$ and $k = 1$.

(b) $\mathbb{E}[R]$ versus the occupancy for different values of d, k with $d/k = 2$.

Fig. 1. Numerical examples: Red(d, k, δ)

Proposition 6.2. *For the Red(d, k, δ) policy, the FDE in equation (3) reduces to the following DIDE (recall $\bar{F}_{R_x}$ and $\bar{F}_{R_{\tilde{X}}}$ from (6-7)):*

$$\bar{F}'(w) = -\lambda d(\bar{F}_{R_{\tilde{X}}}(w) - \bar{F}(w)) \qquad \text{if } w \le \delta \tag{8}$$

$$\bar{F}'(w) = -\lambda d\Bigg(\int_0^\infty \sum_{j=0}^{k-1} \binom{d-1}{j} F_{R_x}(w-\delta)^j \bar{F}_{R_x}(w-\delta)^{d-j-1} \cdot \big(\bar{F}_{R_x}(w) - \bar{F}(w)\big)\, f_X(x)\, dx \Bigg) \qquad \text{otherwise.} \tag{9}$$

Proof. The proof is given in Appendix A.5. □

Remark. *In the special case $d = k$, this policy reduces to the classic fork-join policy and one finds that $\sum_{j=0}^{d-1} \binom{d-1}{j} F_{R_x}(w-\delta)^j \bar{F}_{R_x}(w-\delta)^{d-j-1} = 1$. Therefore we simply have $\bar{F}'(w) = -\lambda d(\bar{F}_{R_{\tilde{X}}}(w) - \bar{F}(w))$.*

Remark. *Taking $\delta = 0, k = 1, X \overset{d}{=} 1$ and $g(S, X) = SX = S$, we find that $\bar{F}$ satisfies:*

$$\bar{F}'(w) = -\lambda d(\bar{F}_{R_1}(w) - \bar{F}(w))\bar{F}_{R_1}(w)^{d-1}$$

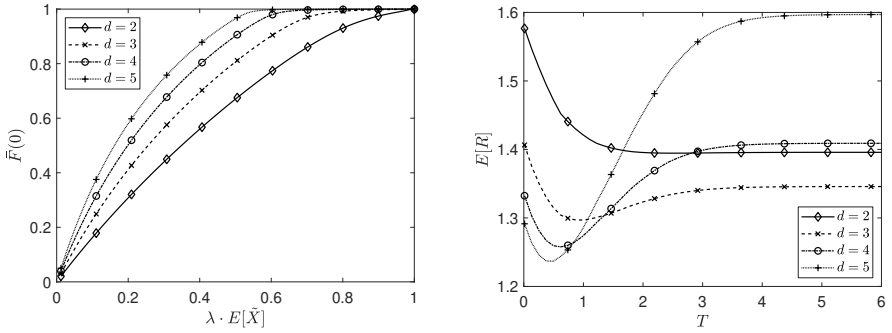
$$\bar{F}_{R_1} = \bar{F}_S(w) + \int_0^w \bar{F}(w-u) f_{S_c}(u)\, du + \sum_u \bar{F}(w-u)\mathbb{P}\{S_d = u\}.$$

It is not hard to see that these equations correspond to (20-21) in [9], this shows how previous work on Red(d) with i.i.d. replicas fits into our framework. Furthermore, Appendix A.13.1 indicates how Theorem 3 from [10] for the case of identical job sizes can be obtained by focusing on the case with no slowdown (i.e., $g(S, X) = X$).

Corollary 6.3. *For the Red(d, k, δ) policy, the ccdf of the equilibrium response time distribution for the queue at the cavity is given by:*

$$\bar{F}_R(w) = \int_0^\infty \left(\sum_{j=0}^{k-1} \binom{d}{j} F_{R_x}(w)^j \bar{F}_{R_x}(w)^{d-j} \right) f_X(x)\, dx.$$

Proof. This follows from the fact that a job is finished as soon as its k'th replica finishes. This time is given by the k'th order statistic of $\{U_i + g(S_i, X)\}$. □



(a) $\bar{F}(0)$ versus $\lambda\mathbb{E}[\tilde{X}]$ for different d

(b) $\mathbb{E}[R]$ versus T for different d

Fig. 2. Numerical examples: RTQ(d, T)

Numerical examples

We take $g(s, x) = (s + 1)x$, X geometric with parameter $1/2$ scaled down such that $\mathbb{E}[X] = 1$ and set S equal to zero with probability $1 - q$ and some other distribution with mean one with probability q. In Figure 1a, we consider $d = 2, k = 1, \delta = 0.01$ and plot the stability region, i.e., $\lambda_{max}\mathbb{E}[\tilde{X}]$, as a function of the slowdown probability q. Note that this value is constant and equal to one without replication (i.e., when $k = d = 1$). Here $\lambda_{\max}$ represents the value such that the system is stable for all $\lambda < \lambda_{\max}$, but unstable for $\lambda \geq \lambda_{\max}$. As explained in Section 8.2 $\lambda_{\max}$ is found by looking at the smallest λ such that the FDE no longer has a proper solution.

We considered different slowdown distributions (when the slowdown is non-zero) namely, Erlang with 2 phases, mean 1 and SCV $1/2$, exponential with mean one, and Hyperexponential with balanced means, mean one and SCV 2 and 3. We observe in Figure 1a that, while for $q = 0$ the value $\lambda_{max}\mathbb{E}[\tilde{X}]$ is evidently the same for different slowdown distributions (as there is no slowdown), a slowdown with a higher coefficient of variation has larger values of $\lambda_{max}\mathbb{E}[\tilde{X}]$ for any $q \in (0, 1]$. Thus there is a more substantial risk to replicate if the slowdown is less variable. One perhaps surprising observation is the fact that $\lambda_{max}\mathbb{E}[\tilde{X}]$ is not monotone, and an optimum value of q emerges.

In Figure 1b we plot $\mathbb{E}[R]$ as a function of λ for different values of d and k while keeping the amount of redundancy fixed to $d/k = 2$. We set $k = 1, 2, 3$ (and consequently $d = 2, 4, 6$) and $q = 0.2$. We assume that the job size is geometric with unit mean (as before) and the slowdown is exponential with unit mean. We observe that increasing the number of parts we divide the job into generally decreases the mean response time, but the value of $\lambda_{\max}$ decreases slightly as k increases. See Section 8 for more details on how to obtain $\bar{F}$ and $\lambda_{\max}$.

6.2 Type 1 : RTQ(d, T)

In this section we look at the RTQ(d, T) policy (redundant to threshold queue). For this policy, we select d queues and replicate on all queues which have a workload of at most T (or assign the job randomly in case all selected queues have a workload which exceeds T). As soon as one replica finishes service, the others are cancelled. Such a scheme is useful in situations where the communication overhead is costly as a server only needs to send a signal to the dispatcher at the time of upcrossing or downcrossing of the threshold T. Note that the Replicate on Idle Queue RIQ(d) policy studied in [8] is a special case of this policy when $T = 0$. A policy that was studied by simulation in [8] is THRESHOLD-n where incoming jobs are replicated on servers with at most

n jobs. RTQ(d, T) can be seen as a workload equivalent of this policy. We discuss how this policy can be implemented in a real system without using any knowledge about the work at the servers in Appendix A.3.2.

As for Red(d, k, δ), we again find that (3) reduces to a DIDE without boundary condition.

Proposition 6.4. *For the RTQ(d, T), the FDE (3) reduces to the following DIDE:*

$$\bar{F}'(w) = -\lambda d \int_0^\infty \bar{F}_{R_x}(w)^{d-1}(\bar{F}_{R_x}(w) - \bar{F}(w)) f_X(x)\, dx \qquad w \le T \tag{10}$$

$$\bar{F}'(w) = -\lambda d \int_0^\infty \left(B_x(w,T)(\bar{F}(T) + B_x(w,T))^{d-1}\right) f_X(x)\, dx$$
$$- \lambda \int_0^\infty (\bar{F}_{R_x}(w) - \bar{F}(w) - B_x(w,T))\bar{F}(T)^{d-1} f_X(x)\, dx \qquad w > T. \tag{11}$$

with:

$$B_x(w,T) = \bar{F}_{g_x(S)}(w)F(T) + \int_0^T f_{g_x(S)}(w-u)(\bar{F}(u) - \bar{F}(T))\, du.$$

Here $B_x(w,T)$ is the probability that an arrival of size x to a single queue increases its workload from somewhere below T to a value above w.

Proof. The proof can be found in Appendix A.6. □

From the equilibrium workload distribution, we obtain the response time distribution:

Corollary 6.5. *For the RTQ(d, T) policy, the ccdf of the equilibrium response time distribution for the queue at the cavity is given by:*

$$\bar{F}_R(w) = \bar{F}(T)^d + \int_0^\infty \left(\sum_{k=1}^{d} \binom{d}{k} \bar{F}(T)^{d-k} B_x(w,T)^k\right) f_X(x)\, dx \qquad w \le T$$

$$\bar{F}_R(w) = \int_0^\infty \left[\bar{F}(T)^{d-1}(\bar{F}_{R_x}(w) - B_x(w,T)) + \sum_{k=1}^{d} \binom{d}{k} \bar{F}(T)^{d-k} B_x(w,T)^k\right] f_X(x)\, dx \qquad w > T$$

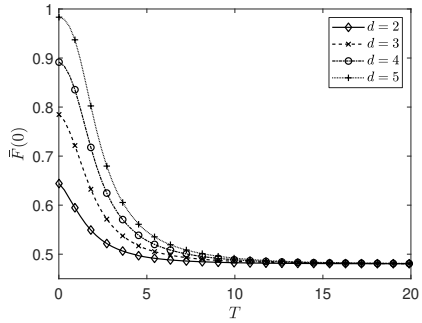
where $B_x(w,T)$ is defined as in Proposition 6.4.

Proof. The proof can be found in Appendix A.7. □

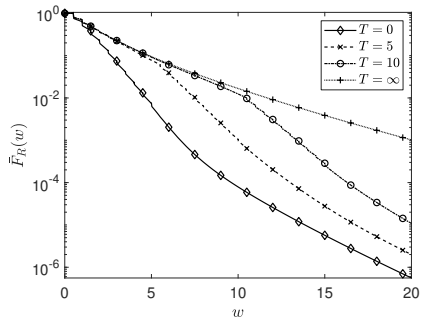
Numerical examples

We now consider some numerical examples for the RTQ(d, T) policy. We set $d = 2, 3, 4, 5$, assume a scaled geometric job size distribution X as for Red(d, k, δ), the probability of a slowdown equals $q = 0.2$ and the slowdown distribution is exponential with unit mean. We take $g(s, x) = (1 + s)x$ and recall that $\tilde{X} = g(S, X)$. In Figure 2a, we take $T = 3$ and show the load of the system given by $\bar{F}(0)$ as a function of the arrival rate for various values of d. We observe that the load $\bar{F}(0)$ increases with d and may be close to one for moderate values of $\lambda\mathbb{E}[\tilde{X}]$. Nevertheless the system remains stable as long as $\lambda\mathbb{E}[\tilde{X}] < 1$ since we never replicate on queues with a workload exceeding T. Understanding the actual system load may be of interest with respect to the energy usage of the servers.

In Figure 2b, we consider the same setting, but fix $\lambda = 0.4$ and show the mean response time for the system as a function of the threshold T. We note that $\mathbb{E}[R]$ stabilizes as T increases. This is due to the fact that for sufficiently large T the workload at all sampled queues is below the threshold and the system behaves nearly identical to the Red$(d, 1, 0)$ policy. For $d = 2$ we observe that the mean response time $\mathbb{E}[R]$ decreases monotonically with T, this is due to the fact that the load is



(a) The system load as a function of T



(b) Tail plot of the ccdf.

Fig. 3. Numerical examples: DR(d, T)

sufficiently low such that it is optimal to replicate all incoming jobs. For higher values of d, we notice that $\mathbb{E}[R]$ initially decreases and subsequently increases. Remarkably we observe that $d = 5$ yields the lowest mean response time by choosing the optimal T but also has the worst performance when we pick T too large. Further note that the optimal value of T decreases in d.

6.3 Type 2 : DR(d, T)

We now analyse the Delayed Replication policy: DR(d, T). This policy has a Type 2 FDE because, as we shall see shortly, when rewriting (3) the right hand side depends on $\bar{F}(u)$ for $u > w$. With this load balancing policy, a job is sent to an arbitrary server (which we call the primary server) on arrival. If the server has not finished service of this job within some fixed time $T \geq 0$, the job is replicated on $d-1$ other servers that are chosen uniformly at random. A cancellation-on-completion policy is then employed on these d replicas of the same job. Note that for $T = 0$ this policy reduces to Red(d) while for $T = \infty$, it reduces to random assignment. We obtain the following result:

Proposition 6.6. *For the DR(d, T), the FDE (3) reduces to the following FDE:*

$$\bar{F}'(w) = -\lambda \int_0^\infty (\bar{F}_{R_x}(w) - \bar{F}(w))\Big((d-1)\bar{F}_{R_x}(w)^{d-2}\bar{F}_{R_x}(w+T) + 1\Big) f_X(x)\,dx \qquad w \leq T \tag{12}$$

$$\bar{F}'(w) = -\lambda \int_0^\infty \big(\bar{F}_{R_x}(w) - \bar{F}(w)\big)\left((d-1)\bar{F}_{R_x}(w)^{d-2}\bar{F}_{R_x}(w+T) + \bar{F}_{R_x}(w-T)^{d-1}\right) f_X(x)\,dx \qquad w > T \tag{13}$$

Proof. The proof can be found in Appendix A.8. □

Remark. *For $T = \infty$, $\bar{F}_{R_x}(w+T) = 0$ and (12) reduces to: $\bar{F}'(w) = -\lambda(\bar{F}_R(w) - \bar{F}(w))$. It is not hard to see that this indeed corresponds to the DIDE for the random assignment policy. On the other hand, for $T = 0$, (13) reduces to $\bar{F}'(w) = -\lambda d \int_0^\infty f_X(x)(\bar{F}_{R_x}(w) - \bar{F}(w))\bar{F}_{R_x}(w)^{d-1}\,dx$, which indeed corresponds to the Red($d, 1, 0$) policy.*

Corollary 6.7. *For the DR(d, T) policy, the ccdf of the equilibrium response time distribution for the queue at the cavity is given by:*

$$\bar{F}_R(w) = \bar{F}_{R_{\hat{X}}}(w) \qquad w \leq T$$

$$\bar{F}_R(w) = \int_0^\infty \bar{F}_{R_x}(w)\bar{F}_{R_x}(w-T)^{d-1} f_X(x)\,dx \qquad w > T.$$

Proof. We make a distinction between the two cases $w > T$ and $w \leq T$. For $w \leq T$, after time w has elapsed, the job is still only being run on one server, thus the probability that it is still running after that time is the same as for random routing with job size $\tilde{X}$. For $w > T$ we find that $d-1$ other servers start serving the job after time T. As they have a backlog of time T, they should finish the job in time $w - T$ to respond before time w. □

Numerical examples

We again take the probability of a slowdown $q = \mathbb{P}\{S > 0\} = 0.2$, S exponential (if it is non-zero), X geometric and assume that $\lambda = 0.4$. In Figure 3a we plot $\bar{F}(0)$ as a function of T. For all values of d, we observe that the system load decreases monotonically as a function of T and converges to $\lambda \cdot \mathbb{E}[\tilde{X}] = 0.48$ as T tends to infinity, which is the load for random routing (as expected).

In Figure 3b, we plot the ccdf of the response time $\bar{F}_R(w)$ as a function of T for different values of d. We observe that initially the response time distribution is close to that of random routing, but once w passes T it quickly starts falling off in a similar fashion as in Red(d), i.e., when $T = 0$. Taking another look at Figure 3a we observe that when setting $T \geq 5$ delayed replication can mitigate so-called stragglers while only slightly increasing the load.

6.4 Type X : Replicate only small jobs

In replication based policies such as Red(d, k, δ), an incoming job is replicated on all the d sampled servers. A drawback of this approach is that the stability region of the policy is typically reduced due to the added work arising from the replicas. Therefore from a stability point of view, one may wish to replicate jobs in a selective manner. One possible alternative, also considered in [8] for RIQ, is to only replicate small jobs. For example, if we have some replication based policy (say policy 1) and another policy which does not use replication (say policy 2), then one can design a new policy where we set some threshold $\tilde{x} \in [0, \infty)$ and assign jobs with inherent job size $x \leq \tilde{x}$ as per policy 1 and the remaining jobs using policy 2. To rewrite equation (3) for such a generic policy (say policy 3), we assume a job of size X samples d queues on arrival where it experiences slowdown $S_1, \ldots, S_d$ respectively and where the workloads are $U_1, U_2, \ldots, U_d$. For $i = 1, 2, 3$, let $\boldsymbol{Q}_i(U_1, U_2, \ldots, U_d, S_1, \ldots, S_d, X)$, denote the new workload in the queue at cavity after a potential arrival occurs in a system with policy i. One finds:

$$\begin{aligned}
&\mathbb{P}\{\boldsymbol{Q}_3(U_1, U_2, \ldots, U_d, S_1, \ldots, S_d, X) > w, U_1 \leq w\} \\
&= \int_0^{\tilde{x}} \mathbb{P}\{\boldsymbol{Q}_1(U_1, U_2, \ldots, U_d, S_1, \ldots, S_d, x) > w, U_1 \leq w\} f_X(x)\, dx \\
&+ \int_{\tilde{x}+}^{\infty} \mathbb{P}\{\boldsymbol{Q}_2(U_1, U_2, \ldots, U_d, S_1, \ldots, S_d, x) > w, U_1 \leq w\} f_X(x)\, dx, \qquad (14)
\end{aligned}$$

where $\int_{\tilde{x}+}^{\infty}$ denotes the integral starting in $\tilde{x}$ excluding this value. We can also easily compute the response time distribution for policy 3. Let $R^{(i)}$ denote the response time of a job sent using policy i for $i = 1, 2, 3$ in a system which employs policy 3. The ccdf $\bar{F}_{R^{(i)}}$, $i = 1, 2$ can be computed in the same manner as for policy i, but now using the workload distribution of policy 3. The response time of a general job is then found as:

$$\bar{F}_{R^{(3)}}(w) = \mathbb{P}\{X \leq \tilde{x}\} \bar{F}_{R^{(1)}}(w) + \mathbb{P}\{X > \tilde{x}\} \bar{F}_{R^{(2)}}(w). \qquad (15)$$

As a simple example, we now analyse a policy which applies Red$(d, 1, 0)$ whenever the inherent job size $X \leq \tilde{x}$ and random assignment otherwise. It is not hard to see that (14) simplifies to:

$$\bar{F}'(w) = -\lambda d \int_0^{\tilde{x}} (\bar{F}_{R_x}(w) - \bar{F}(w))\bar{F}_{R_x}(w)^{d-1} f_X(x)\, dx - \lambda \int_{\tilde{x}+}^{\infty} (\bar{F}_{R_x}(w) - \bar{F}(w)) f_X(x)\, dx.$$

Note that this is still a Type 1 FDE, which can be analysed in the exact same way as the regular Red$(d, 1, 0)$ policy (c.f. Section 8). Moreover, letting $X_1 = (X \mid X \leq \tilde{x})$ and $X_2 = (X \mid X > \tilde{x})$, we find that the response time distributions are given by:

$$\bar{F}_{R^{(1)}}(w) = \int_0^{\tilde{x}} \bar{F}_{R_x}(w)^d f_{X_1}(x)\, dx$$

for the jobs which are replicated and

$$\bar{F}_{R^{(2)}}(w) = \bar{F}_{g(S,X_2)}(w) + \int_0^w f_{g(S,X_2)}(w-u)\bar{F}(u)\, du$$

for the randomly routed jobs. One can employ (15) to obtain the response time distribution for an arbitrary job. There is of course nothing special about this choice for policies 1 and 2, and this approach can be employed to combine any two arbitrary policies.

6.5 Type 3 : LL(d, k, δ)

For the Least Loaded LL(d, k, δ) policy, when a job consisting of k equally sized parts arrives, the dispatcher sends a placeholder for this job to $d \geq k$ FCFS servers that are sampled uniformly at random. When a placeholder reaches the head of a queue, the server informs the dispatcher and the dispatcher assigns one of the k parts of the job to the server as long as at least one part remains. We assume that the server requires a time $\delta \geq 0$ to inform the dispatcher (and to potentially receive the part). Note that this request mechanism corresponds to the late binding mechanism used in [17]. In what follows, we first analyse the LL(d, k)=LL$(d, k, 0)$ policy. This is followed by the more general analysis for LL(d, k, δ), $\delta \geq 0$. More discussion on the LL(d, k) policy can be found in Appendix A.3.3.

For the LL(d, k) policy we denote $\boldsymbol{Q}(U_1) = \boldsymbol{Q}(U_1, \ldots, U_d, S_1, \ldots, S_d, X)$ for the workload of the cavity queue after a potential arrival of size $k \cdot X$ occurs, where the d servers have workloads $U_1, \ldots, U_d$ and experience slowdowns $S_1, \ldots, S_d$. In case $U_1 > 0$, $\boldsymbol{Q}(U_1)$ is equal in distribution to $U_1 + \tilde{X}$ if U_1 is one of the k least loaded servers among $U_1, \ldots, U_d$ and U_1 otherwise (recall that $\tilde{X} = g(S, X)$). In case $U_1 = 0$, $\boldsymbol{Q}(U_1)$ equals in distribution $\tilde{X}$ with probability $\min\left\{1, \frac{k}{|\{i|U_i=0\}|}\right\}$ and 0 otherwise.

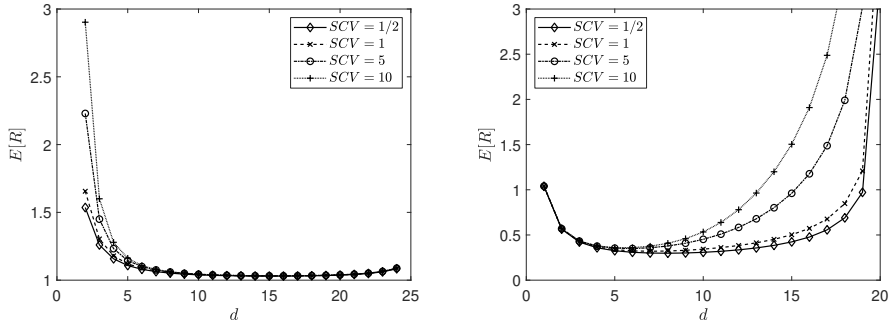
Let $\rho = k\lambda\mathbb{E}[\tilde{X}]$ denote the amount of work that one arrival creates. Note that a system operating under the LL(d, k) policy is stable iff $\rho < 1$. In the following proposition, we derive a DIDE with boundary condition which characterizes the equilibrium workload distribution:

Proposition 6.8. *For the LL(d, k, δ), the FDE (3) reduces to the following DIDE:*

$$\bar{F}'(w) = -\lambda\left(\bar{F}_{\tilde{X}}(w) + H(w) - \int_0^w H(u) f_{\tilde{X}}(w-u)\, du\right), \tag{16}$$

with:

$$H(w) = \sum_{j=1}^{d} \min\{j, k\}\binom{d}{j}\bar{F}(w)^{d-j}(1-\bar{F}(w))^j - 1 \tag{17}$$



(a) $k = 1$ fixed and $d = 2, 3, \ldots, 24$. (b) $d = 20$ fixed and $k = 1, 2, \ldots, d$.

Fig. 4. Numerical examples: LL(d, k, δ)

and $\bar{F}(0) = \rho$. Equivalently, this DIDE can be written as a fixed point equation (FPE):

$$\bar{F}(w) = \rho - \lambda \int_0^w (1 + H(u))\bar{F}_{\tilde{X}}(w - u)\, du. \tag{18}$$

Proof. The proof can be found in Appendix A.9. □

Remark. *In case $k = 1$, we find that $H(w) = -\bar{F}(w)^d$, and (18) reduces to the fixed point equation that was obtained in [11]. In particular, (16) yields an alternative method to compute the ccdf of the workload distribution in case job sizes are not PH distributed.*

Remark. *In case $\tilde{X}$ is exponential, one can show that (18) is equivalent to a simple ODE:*

$$\bar{F}'(w) = \rho - \bar{F}(w) - \lambda(1 + H(w)). \tag{19}$$

The proof goes along the same lines as the proof of Theorem 5.1 in [11], unfortunately for $2 \leq k < d$ this ODE does not have a simple closed form solution.

We define ccdf_ρ as the set of all ccdfs which start in ρ, i.e. functions on $[0, \rho]^{[0,\infty)}$ which satisfy: $\bar{F}(0) = \rho$, $\lim_{w\to\infty} \bar{F}(w) = 0$, for all $w, s > 0 : \bar{F}(w + s) \leq \bar{F}(w)$ and $\lim_{s\to 0^+} \bar{F}(w + s) = \bar{F}(w)$. We can show the following:

Proposition 6.9. *If we let $T_d^{(k)} : \text{ccdf}_\rho \to [0, 1]^{[0,\infty)}$ be defined by: $T_d^{(k)}\bar{F}(w) = \rho - \lambda \int_0^w (1 + H(u))\bar{F}_{\tilde{X}}(w - u)\, du$. Then we have $T_d^{(k)}\bar{F} \in \text{ccdf}_\rho$ for all $\bar{F} \in \text{ccdf}_\rho$. Moreover for $\bar{F}_1, \bar{F}_2 \in \text{ccdf}_\rho$ we have (with d_K the Kolmogorov distance):*

$$d_K(T_d^{(k)}\bar{F}_1, T_d^{(k)}\bar{F}_2) < A(d, k, \lambda) d_K(\bar{F}_1, \bar{F}_2), \tag{20}$$

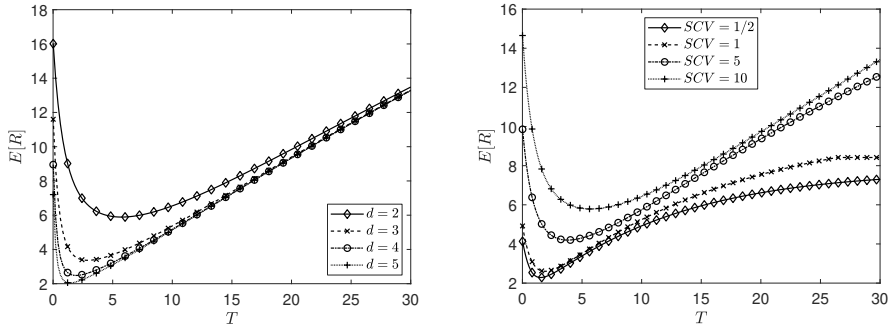
with:

$$A(d, k, \lambda) = \sup_{x\in[\rho,1]} \left(d \sum_{j=1}^{k-1} (k - j)(1 - x)^{d-j-1} x^j \right) \leq dk^2 \lambda^{d-k} \underset{d\to\infty}{\longrightarrow} 0.$$

Proof. This proof can be found in Appendix A.10. □

Remark. *As (16) and (18) are equivalent, we find (from the Banach-fixed point theorem) that all these equations have a unique solution provided that $dk^2\rho^{d-k} < 1$ (with boundary condition $\bar{F}(0) = \rho$).*

Based on our analysis for LL(d, k) we now find the following result for the more general setting with arbitrary δ.



(a) $\mathbb{E}[R]$ versus T for various values of d (b) $\mathbb{E}[R]$ versus T for different distributions of $\tilde{X}$.

Fig. 5. Numerical examples: JTQ(d, T)

Proposition 6.10. *For the LL(d, k, δ), the FDE (3) reduces to the following DIDE:*

$$\bar{F}'(w) = -\lambda d(1 - \bar{F}(w)) \qquad w \leq \delta$$

$$\bar{F}'(w) = -\lambda d(\bar{F}(w-\delta) - \bar{F}(w)) - \lambda\left(\bar{F}_{\tilde{X}}(w-\delta) + H(w-\delta) - \int_0^{w-\delta} H(u) f_{\tilde{X}}(w-\delta-u)\, du\right) \qquad w \geq \delta,$$

with boundary condition $\bar{F}(0) = \lambda(\mathbb{E}[\tilde{X}] + d\delta)$

Proof. This easily follows by applying the fact that $Q(U_1) = Q_{LL(d,k)}(U_1) + \delta$, where $Q_{LL(d,k)}$ is defined as the $Q-$function corresponding to the LL(d, k) policy. The boundary condition follows from the fact that each job brings $\mathbb{E}[\tilde{X}] + d\delta$ work on average. □

If there is no slowdown (i.e. $g(s, x) = x$) and jobs are split into k identical parts, then it is obvious that the response time of a job is given by its response time at the server with the k'th lowest workload for which the ccdf is easy to compute. When computing the response time for the LL(d, k, δ) policy in the *S&X* model, things are more involved, see Appendix A.11.

Numerical examples

We investigate how the choice of d and k impact the mean response time in the LL(d, k, δ) policy described above in case there is no slowdown and each incoming job is split into k equal parts. We fix $\lambda = 0.8$ and $\delta = 0.01$, for the jobs we take Erlang distributed job sizes with mean 1 and SCV 1/2, exponential job sizes with mean one, and Hyperexponential job sizes with balanced means, unit mean and SCV 5 resp. 10.

In Figure 4a, we observe that increasing the value of d decreases the mean response time where the gain is greater for the more variable jobs. As d becomes large, the mean response times seem to coincide for all values of the SCV. For d sufficiently large the mean response time starts to increase due to the extra load coming from the request time δ. Note that the system becomes unstable for $d = 25$ as the system load is $0.8 + 0.01 \cdot 25 \cdot 0.8 = 1$. In Figure 4b, we observe that splitting a job into multiple parts initially yields a gain in response times, even driving response times below 1. As we divide jobs into more and more parts, we observe that the response times start to increase as the number of servers we can choose decreases. Note that there is a very strong increase when going from $k = 19 = d - 1$ to $k = 20 = d$ jobs, this is due to the fact that at $k = d$ we completely loose the power of d choices.

6.6 Type 4 : JTQ(d, T)

In the JTQ(d, T) policy, on arrival of a job, d servers are sampled and the job is served on a randomly chosen server (among the d servers) which has a workload below the threshold T. In case none of the sampled d servers has a workload of at most T, the job is randomly routed to one of the d servers. When $T = \infty$, this reduces to random routing. Like the RTQ(d, T) policy, JTQ(d, T) is a policy that has low communication overhead as the servers need to only inform the dispatcher about their level crossings of the threshold T.

We can use the following proposition to compute its equilibrium workload distribution.

Proposition 6.11. *For the JTQ(d, T), the FDE (3) reduces to the following FDE:*

$$\bar{F}'(w) = -\lambda\left(\frac{1-\bar{F}(T)^d}{1-\bar{F}(T)}(\bar{F}_{R_{\tilde{X}}}(w) - \bar{F}(w))\right) \qquad w \le T$$

$$\bar{F}'(w) = -\lambda\left(\frac{1-\bar{F}(T)^d}{1-\bar{F}(T)}A(w,T) + \bar{F}(T)^{d-1}\left[\bar{F}_{R_{\tilde{X}}}(w) - \bar{F}(w) - A(w,T)\right]\right) \qquad w > T,$$

with $A(w,T) = \bar{F}_{\tilde{X}}(w)F(T) + \int_{w-T}^{w} f_{\tilde{X}}(u)(\bar{F}(w-u) - \bar{F}(T))\,du$ and $R_{\tilde{X}}$ the marginal response time (see Section 4). Here we have the boundary condition $\bar{F}(0) = \lambda\mathbb{E}[\tilde{X}]$.

Proof. The proof can be found in Appendix A.12 □

Remark. *For numerical considerations we may define $H(w) = \bar{F}(w)\delta_{\{w\le T\}}$, we find that $A(w,T) = \bar{G}(w) + \int_0^w g(w-u)H(u)\,du$ which can be quickly computed as it is a convolution. Here $\delta_{\{w\le T\}}$ is one when $w \le T$ and zero otherwise.*

Corollary 6.12. *For the JTQ(d, T) policy, the ccdf of the equilibrium response time distribution for the queue at the cavity is given by:*

$$\bar{F}_R(w) = \frac{1-\bar{F}(T)^d}{1-\bar{F}(T)}\left(\bar{F}_{\tilde{X}}(w)F(T) + \int_0^T f_{\tilde{X}}(w-u)(\bar{F}(u) - \bar{F}(T))\,du\right)$$
$$+ \bar{F}(T)^{d-1}\left(\bar{F}_{\tilde{X}}(w-T)\bar{F}(T) + \int_0^{w-T} f_{\tilde{X}}(w-T-u)\bar{F}(T+u)\,du.\right)$$

Proof. This follows from the fact that $\bar{F}_R(w)$ is given by:

$$(1-\bar{F}(T)^d)\mathbb{P}\{U + \tilde{X} > w \mid U \le T\} + \bar{F}(T)^d\mathbb{P}\{U + \tilde{X} > w \mid U > T\}.$$

□

Numerical examples

We now consider some numerical examples for the JTQ(d, T) policy without slowdown. In Figure 5a, we compare the mean response time $\mathbb{E}[R]$ as a function of T for different values of d. We assume that $\lambda = 0.9$ and $\tilde{X}$ is exponential with unit mean. Firstly, note that $T = 0$ corresponds to JIQ(d). On the other hand, $T = \infty$ corresponds to the random routing policy. We see that for different values of d, $\mathbb{E}[R]$ first decreases and then increases to the same value (due to the resemblance to random routing policy for large T) indicating that the parameter T should be chosen suitably. Furthermore, the optimal value of T decreases with increase in d. Note that as d tends to infinity, JIQ(d) becomes an optimal policy as expected.

In Figure 5b, we show $\mathbb{E}[R]$ versus T with different distributions for $\tilde{X}$: Erlang with mean 1 and SCV 1/2, exponential with mean one, and Hyperexponential with balanced means, mean one and

SCV 5 and 10. We again assume $\lambda = 0.9$ and $d = 2$. For the different distributions, we notice that $\mathbb{E}[R]$ again has the same shape as in Figure 5a. Furthermore, the optimal value of T increases with the SCV.

7 PHASE TYPE DISTRIBUTION

While the analysis presented till now assumes that the job sizes and slowdown variables are generally distributed, in this section, we focus on the case where certain random variables have a phase type (PH) distributions and for all x, $g_x^{-1}(w)$ is linear in w. Note that any distribution on $[0, \infty)$ can be approximated arbitrarily close with a PH distribution, moreover there are various tools available online for matching PH distributions (see e.g. [14], [18]).

A PH distribution with pdf $b(\cdot)$ and cdf $B(\cdot)$ with $B(0) = 0$ is fully characterized by a stochastic vector $\alpha \in \mathbb{R}^n$ and a subgenerator matrix $A \in \mathbb{R}^{n\times n}$ such that $\bar{B}(w) = \alpha e^{Aw}\mathbf{1}$ and $b(w) = \alpha e^{Aw}\mu$ with $n \in \mathbb{N}$, $\mu = -A\mathbf{1}$ and $\mathbf{1}$ an $n \times 1$ column vector consisting of ones. We note that for the choice of $g(s, x) = (s + 1)x$ that we consider in our numerical examples, and the choice of $g(s, x) = sx$ as considered in [8], $g_x^{-1}(w)$ is indeed linear in w.

The importance of the results in this section mostly relate to the computation time of the numerical solution methods. The idea is to replace integral equations with a system of differential equations which are numerically less cumbersome to compute. If we let M denote the number of control points used to define $\bar{F}$, the results involving PH distributions generally reduce computational complexity by one order. For example, we find that solving the DIDE given in Proposition 6.8 with a discrete job size distribution X requires $O(M^2)$ time, whilst the DDE given by Proposition 7.3 can be solved in $O(M)$ time.

7.1 Red(d, k, δ)

For the Red(d, k, δ) policy, the integral is hidden in $\bar{F}_{R_x}(w)$ (see (6)). We show that a simplification in the analysis is possible for obtaining $\bar{F}_{R_x}$. Note that this speed-up applies to all policies which use $\bar{F}_{R_x}$ in their associated FDE:

Proposition 7.1. *Assume $g_x^{-1}(w)$ is linear in w. Let $a(x) = \frac{\partial (g_x)^{-1}}{\partial w}$ and assume S is PH distributed with parameters (α, A). We find:*

$$\begin{aligned}
\bar{F}_{R_x}(w) &= \bar{F}_S(g_x^{-1}(w)) + \alpha a(x)\xi_x(w) \\
\xi_x'(w) &= \bar{F}(w - g_x(0))\mu + A\xi_x(w)a(x), && w > g_x(0) \\
\xi_x(w) &= 0 && w \le g_x(0),
\end{aligned}$$

with $\mu = -A\mathbf{1}$.

Proof. As S is PH distributed, we find:

$$\bar{F}_{R_x}(w) = \bar{F}_S(g_x^{-1}(w)) + \alpha \int_{g_x(0)}^{w} \bar{F}(w - u)e^{g_x^{-1}(u)A}\mu \, du a(x).$$

The result follows by letting $\xi_x(w) = \int_{g_x(0)}^{w} \bar{F}(w - u)e^{g_x^{-1}(u)A}\mu \, du$. Indeed, we first use a substitution to write it as: $\xi_x(w) = \int_{g_x(0)}^{w} \bar{F}(u)e^{g_x^{-1}(w-u)A}\mu \, du$, one may then simply apply the Leibniz integral rule to differentiate $\xi_x(w)$. □

Remark. *Using the result in Proposition 7.1 we can rewrite the DIDE given in (8-9) as:*

$$\bar{F}'(w) = -\lambda d\left[\int_0^\infty \left(\bar{F}_S(g_x^{-1}(w)) + \alpha a(x)\xi_x(w)\right) f_X(x)\,dx - \bar{F}(w)\right] \qquad w \le \delta$$

$$\bar{F}'(w) = -\lambda d\Bigg(\int_0^\infty \sum_{j=0}^{k-1} \binom{d-1}{j} \left(1 - \bar{F}_S(g_x^{-1}(w-\delta)) + \alpha a(x)\xi_x(w-\delta)\right)^j \left(\bar{F}_S(g_x^{-1}(w-\delta))\right.$$
$$\left. + \alpha a(x)\xi_x(w-\delta)\right)^{d-j-1} \cdot \left(\bar{F}_S(g_x^{-1}(w)) + \alpha a(x)\xi_x(w) - \bar{F}(w)\right) f_X(x)\,dx\Bigg) \qquad w > \delta$$

$$\xi_x(w) = 0 \qquad w \le g_x(0)$$

$$\xi_x'(w) = \bar{F}(w - g_x(0))\mu + A\xi_x(w)a(x), \qquad w > g_x(0).$$

It is not hard to see how to generalize this result in case S is a combination of a discrete and a PH distributed random variable. In our numerical examples, we assumed that S is PH distributed with probability q and zero with probability $1-q$ and $g(s,x) = (s+1)x$. Let us denote $\mathcal{A} = (S \mid S > 0)$ the PH distribution S has with probability q, assume it has parameters (α, A) and let $\mu = -A\mathbf{1}$. It is not hard (See also the proof of Proposition 7.2) to show that:

$$\bar{F}_{R_x}(w) = 1 \qquad \text{if } w \le x$$

$$\bar{F}_{R_x}(w) = q\left(\bar{F}_{\mathcal{A}}\left(\frac{w-x}{x}\right) + \frac{\alpha\xi_x(w)}{x}\right) + (1-q)\bar{F}(w-x) \qquad \text{if } w > x$$

$$\xi_x(w) = 0 \qquad \text{if } w \le x$$

$$\xi_x'(w) = \bar{F}(w-x)\mu + \frac{A\xi_x(w)}{x} \qquad \text{if } w > x.$$

The case of no slowdown and PH distributed job sizes can be found in Appendix A.13.1.

7.2 RTQ(d, T)

For the RTQ(d, T) policy, we have an additional integral for $B_x(w,T)$ besides $\bar{F}_{R_x}(w)$. We show how this integral can be eliminated in case S is a combination of a discrete and a PH distribution:

Proposition 7.2. *If in the setting of Proposition 6.4, S is PH distributed with parameters (α, A), $\mu = -A\mathbf{1}$ with probability q, and zero otherwise and $(g_x^{-1})'(w) = a(x)$ does not depend on w, then:*

$$B_x(w,T) = \bar{F}_S(g_x^{-1}(w))F(T) + q\alpha\varphi_x(w)a(x) + (1-q)(\bar{F}(w - g_x(0)) - \bar{F}(T))I_{[g_x(0),T+g_x(0)]}(w),$$

where $\varphi_x(w)$ satisfies:

$$\varphi_x(w) = 0 \qquad w \le g_x(0)$$

$$\varphi_x'(w) = (\bar{F}(w - g_x(0)) - \bar{F}(T))\mu + A\varphi_x(w)a(x) \qquad g_x(0) < w \le T + g_x(0)$$

$$\varphi_x'(w) = A\varphi_x(w)a(x) \qquad T + g_x(0) < w.$$

Proof. Let $\mathcal{A} = (S \mid S > 0)$, one finds that $\int_0^T f_{g_x(S)}(w-u)(\bar{F}(u) - \bar{F}(T)\,du$ equals:

$$q\int_0^T f_{g_x(\mathcal{A})}(w-u)(\bar{F}(u) - \bar{F}(T))\,du + (1-q)(\bar{F}(w - g_x(0)) - \bar{F}(T))I_{[g_x(0),T+g_x(0)]}(w).$$

Furthermore as $g_x(\mathcal{A}) \geq g_x(0)$ we find that $f_{g_x(\mathcal{A})}(w-u) = 0$ for $w - u < g_x(0)$ which happens if $w - g_x(0) < u$. The result now follows by setting:

$$\varphi_x(w) = \int_0^{\min\{T,(w-g_x(0))\}} e^{g_x^{-1}(w-u)A}(\bar{F}(u) - \bar{F}(T))\, du\mu.$$

□

As for Red(d,k,δ), we obtain an alternative characterization of the equilibrium workload distribution for the RTQ(d,T) policy by combining Proposition 6.4, 7.1 and 7.2. The case of no slowdown and PH distributed job sizes is discussed in Appendix A.13.2.

7.3 LL(d,k,δ)

In this section, we look at the scenario where the actual job size $\tilde{X} = g(S,X)$ is PH distributed. It was already noted in [11] and [10] that for the LL(d) policy, when job sizes are PH distributed, the associated IDE can be reduced to a DDE which can be solved more efficiently. We show a similar result for LL(d,k,δ).

Proposition 7.3. *If $\tilde{X}$ is PH distributed with parameters (α, A) we find that the DIDE given in Proposition 6.10 simplifies to the following DDE:*

$$\begin{aligned}
\bar{F}'(w) &= -\lambda d(1-\bar{F}(w)) && \text{if } w \leq \delta \\
\bar{F}'(w) &= -\lambda d(\bar{F}(w-\delta) - \bar{F}(w)) - \lambda\left(\bar{F}_{\tilde{X}}(w-\delta) + H(w-\delta) - \alpha\xi(w-\delta)\right) && \text{if } w \geq \delta \\
\xi'(w) &= A\xi(w) + H(w)\mu,
\end{aligned}$$

with boundary condition $\xi(0) = 0$ and $\mu = -A\mathbf{1}$.

Proof. This easily follows from Proposition 6.8 by noting that $f_{\tilde{X}}(w) = \alpha e^{wA}\mu$ and setting $\xi(w) = \int_0^w H(u) f_{\tilde{X}}(w-u)\, du$. □

This result can be further generalized in case $\tilde{X}$ is a combination of a discrete and a PH distribution.

8 NUMERICAL METHOD

In this section we discuss the numerical algorithm used to generate the numerical examples for the system stability and workload/response time distribution presented in the paper. As stated earlier, a comparison with results obtained using time consuming simulation experiments is presented in Appendix A.1 and A.2.

8.1 Computing the workload distribution

The equilibrium workload distribution can be obtained from a simple forward Euler scheme for future independent policies (Type 1 and Type 3) as the right hand side of these equations only depends on $\bar{F}(u)$ for $u \leq w$. For future dependent policies (Type 2 and Type 4), we observe that the right hand side also depends on $\bar{F}(u)$ for $u > w$, therefore, one may rely on a Fixed Point Iteration to obtain the equilibrium workload distribution. However, note that Theorem 5.2 does not specify a boundary condition for $\bar{F}(0)$. This is not surprising as $\bar{F}(0)$ corresponds to the actual system load which is unknown for some policies. When this load is known, i.e. for Type 3 and Type 4 systems, we can simply use this system load as a boundary condition, i.e. set $\bar{F}(0) = \lambda(d\delta + \mathbb{E}[\tilde{X}])$ for LL(d,k,δ) and $\bar{F}(0) = \lambda\mathbb{E}[\tilde{X}]$ for JTQ(d,T). However for other policies (mostly those that contain some type of redundancy), $\bar{F}(0)$ is unknown.

We do know that if $\bar{F}$ is the ccdf of a workload distribution, it must satisfy $\inf_{w>0} \bar{F}(w) = 0$. Based on this trivial observation, we obtain an algorithm which can be used to find the solution $\bar{F}$ for (3) when the system is stable (i.e. the equilibrium workload distribution is not infinite) and an algorithm to obtain the highest value of λ for which it is still stable. To this end we first define two simple operators, $T_1 : (0,1) \to \mathbb{R}^{[0,\infty)}$ and $T_2 : \mathbb{R}^{[0,\infty)} \times (0,1) \to \mathbb{R}^{[0,\infty)}$. Here T_1 maps a value x_0 to the solution found by solving the corresponding DIDE with boundary condition $\bar{F}(0) = x_0$, using a forward Euler iteration. To define T_2 we first define:

$$R_{x_0}\bar{F} = x_0 - x_0 \frac{\bar{F}(0) - \bar{F}}{\bar{F}(0)},$$

which scales $\bar{F}$ to satisfy $\bar{F}(0) = x_0$. Secondly, we define $\mathcal{H}_d$ as:

$$\mathcal{H}_d\bar{F}(w) = x_0 - \lambda d \int_0^w \bar{F}'(u)\, du = x_0 - \lambda d \int_0^w \mathbb{P}\{Q(U) > u, U \le u\}\, du.$$

We now let $T_2(\bar{F}, x_0)$ denote the operator which first applies R_{x_0} to $\bar{F}$ and then repeatedly applies $\mathcal{H}_d$ until $\|\bar{F} - \mathcal{H}_d\bar{F}\|_\infty$ is sufficiently small. Using the operators T_1 and T_2, we propose an algorithm to obtain the equilibrium workload distribution. This algorithm is basically a simple bisection algorithm (on T_1 for future independent and T_2 for future dependent policies), where we look for $\bar{F}(0)$ such that $\inf_{w>0} \bar{F}(w) = 0$.

Step 1: Set lb = 0, ub = 1, $n = 0$ and $\bar{F}_0(w) = \bar{F}_{\tilde{X}}(w)$.

Step 2: Set $x_0 = \frac{lb+ub}{2}$ and compute $\bar{F}_{n+1} = T_1(x_0)$ for a future independent resp. $\bar{F}_{n+1} = T_2(\bar{F}_n, x_0)$ for a future dependent policy.

Step 3: Compute $y = \inf_{w>0} \bar{F}_{n+1}(w)$ and increment n by one.

Step 4: Set lb $= x_0$ if $y \le 0$ otherwise set ub $= x_0$, return to Step 2.

Terminate the algorithm when $|\inf_{w>0} \bar{F}(w)|$ is sufficiently small.

Remark. *For the policies where $\bar{F}(0)$ is known, one can simply set* lb = ub $= \bar{F}(0)$ *in step 1 and the equilibrium workload distribution is given by $\bar{F}_1$.*

It is not hard to see that if $\bar{F}$ satisfies any of the future (in)dependent FDEs considered in this work and $\inf_{w\ge 0} \bar{F}(w) = 0$, then $\bar{F}$ is indeed a ccdf. To this end one essentially needs to show that $\bar{F}(w)$ is non-increasing. For example for Red(d, k, δ), one can establish that $\bar{F}_{R_{\tilde{X}}}(w) \ge \bar{F}(w)$ for all w and $\bar{F}_{R_x}(w) \ge \bar{F}(w)$ for all w, x. From this it then follows that $\bar{F}$ is indeed decreasing, the property $\inf_{w\ge 0} \bar{F}(w) = 0$ then ensures that $\bar{F}(w) \ge 0$ and $\lim_{w\to\infty} \bar{F}(w) = 0$.

However, to be certain that this algorithm converges to the equilibrium workload distribution, one needs to show that if $\bar{F}_1$ and $\bar{F}_2$ are two solutions of the same FDE, that satisfy $\bar{F}_1(0) \le \bar{F}_2(0)$ and $\inf_{w>0} \bar{F}_2(w) < 0$ then also $\inf_{w>0} \bar{F}_1(w) < 0$. Proving this seems difficult, but numerical experiments suggest that this is indeed the case for all examples considered. For LL(d, k, δ) with exponential job sizes this trivially holds as the DIDE is equivalent to the ODE (19) (this is also the case for Red(d) with independent replicas and exponential job sizes). Moreover, convergence in our algorithm is not guaranteed (except to some extent for LL(d, k) by Theorem 6.8).

8.2 Stability

We let $\lambda_{\max}$ denote the smallest arrival rate λ for which the load balancing policy is no longer stable, i.e., for all $\lambda < \lambda_{\max}$ stability is ensured while for $\lambda \ge \lambda_{\max}$ the system is unstable.

In order to approximate the unknown value of $\lambda_{\max}$ we need to find the smallest value of λ for which there does not exist any $\bar{F}(0) \in (0,1)$ s.t. the associated solution of Theorem 5.2 satisfies $\inf_{w\ge 0} \bar{F}(w) = 0$, equivalently we must find the smallest value of λ s.t. for each choice of $\bar{F}(0)$ we have $\inf_{w>0} \bar{F}(w) > 0$. In order to approximate this value we pick some sufficiently small $\varepsilon > 0$ and

set $x_0 = 1 - \varepsilon$. We then let $\bar{F} = T_1(x_0)$ resp. $\bar{F} = T_2(\bar{F}_{\tilde{X}}, x_0)$, and check whether $\inf_{w>0} \bar{F}(w) > 0$, if it is, we conclude that the system is (or at least is very close to being) unstable and if $\inf_{w>0} \bar{F}(w) \leq 0$ we conclude that the system is stable. One can thus find an approximation for $\lambda_{\max}$ using a simple bisection method.

9 FUTURE WORK

This paper provides a numerical method to practitioners to assess the performance of workload dependent policies without the need to resort to simulation. In some rare cases, analytical results have been found by solving the FDE obtained in this work (see [11] and [9]). A theoretical follow-up may exist in proving the asymptotic independence for specific policies. Another alley worth investigating is letting d scale with N. A disadvantage of this is that many of the policies become unstable for all $\lambda > 0$ if $d_N \to \infty$. It would also be of interest to generalize these results to the setting of heterogeneous servers, in this case one would need to take a queue at the cavity for each server type. Another interesting application of this method is the case with energy aware servers, where servers shut down when idle and take some time $\delta > 0$ to restart when a job arrives.

REFERENCES

[1] R. Aghajani, X. Li, and K. Ramanan. 2017. The PDE Method for the Analysis of Randomized Load Balancing Networks. *Proc. ACM Meas. Anal. Comput. Syst.* 1, 2, Article 38 (Dec. 2017), 28 pages. https://doi.org/10.1145/3154497

[2] G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica. 2013. Effective Straggler Mitigation: Attack of the Clones.. In *NSDI*, Vol. 13. 185–198.

[3] U Ayesta, T Bodas, JL Dorsman, and IM Verloop. 2019. A token-based central queue with order-independent service rates. *arXiv preprint arXiv:1902.02137* (2019).

[4] U. Ayesta, T. Bodas, and I. M. Verloop. 2018. On a unifying product form framework for redundancy models. *Performance Evaluation* 127 (2018), 93–119.

[5] M. Bramson, Y. Lu, and B. Prabhakar. 2010. Randomized load balancing with general service time distributions. In *ACM SIGMETRICS 2010*. 275–286. https://doi.org/10.1145/1811039.1811071

[6] M. Bramson, Y. Lu, and B. Prabhakar. 2012. Asymptotic independence of queues under randomized load balancing. *Queueing Syst.* 71, 3 (2012), 247–292. https://doi.org/10.1007/s11134-012-9311-0

[7] S. Foss and A. L. Stolyar. 2017. Large-scale join-idle-queue system with general service times. *Journal of Applied Probability* 54, 4 (2017), 995–1007. https://doi.org/10.1017/jpr.2017.49

[8] K. Gardner, M. Harchol-Balter, A. Scheller-Wolf, and B. Van Houdt. 2017. A better model for job redundancy: Decoupling server slowdown and job size. *IEEE/ACM Transactions on Networking* 25, 6 (2017), 3353–3367.

[9] K. Gardner, M. Harchol-Balter, A. Scheller-Wolf, M. Velednitsky, and S. Zbarsky. 2017. Redundancy-d: The Power of d Choices for Redundancy. *Operations Research* 65, 4 (2017), 1078–1094.

[10] T. Hellemans and B. Van Houdt. 2018. Analysis of redundancy (d) with identical replicas. In *Performance evaluation review*. Vol. 46. 74–79.

[11] T. Hellemans and B. Van Houdt. 2018. On the Power-of-d-choices with Least Loaded Server Selection. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 2 (2018), 27.

[12] G. Joshi, Y. Liu, and E. Soljanin. 2012. Coding for fast content download. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE, 326–333.

[13] G. Joshi, E. Soljanin, and G. Wornell. 2017. Efficient redundancy techniques for latency reduction in cloud systems. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)* 2, 2 (2017), 12.

[14] J. Kriege and P. Buchholz. 2014. *PH and MAP Fitting with Aggregated Traffic Traces*. Springer International Publishing, Cham, 1–15. https://doi.org/10.1007/978-3-319-05359-2_1

[15] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. Greenberg. 2011. Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services. *Perform. Eval.* 68 (2011), 1056–1071. Issue 11.

[16] M. Mitzenmacher. 2001. The Power of Two Choices in Randomized Load Balancing. *IEEE Trans. Parallel Distrib. Syst.* 12 (October 2001), 1094–1104. Issue 10.

[17] K. Ousterhout, P. Wendell, M. Zaharia, and I. Stoica. 2013. Sparrow: Distributed, Low Latency Scheduling. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles (SOSP '13)*. ACM, New York, NY, USA, 69–84. https://doi.org/10.1145/2517349.2522716

[18] A. Panchenko and A. Thümmler. 2007. Efficient Phase-type Fitting with Aggregated Traffic Traces. *Perform. Eval.* 64, 7-8 (Aug. 2007), 629–645. https://doi.org/10.1016/j.peva.2006.09.002

[19] N. B Shah, K. Lee, and K. Ramchandran. 2016. When do redundant requests reduce latency? *IEEE Transactions on Communications* 64, 2 (2016), 715–722.
[20] V. Shah, A. Bouillard, and F. Baccelli. 2017. Delay comparison of delivery and coding policies in data clusters. In *Communication, Control, and Computing (Allerton), 2017 55th Annual Allerton Conference on*. IEEE, 397–404.
[21] A. L. Stolyar. 2015. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Systems* 80, 4 (2015), 341–361. https://doi.org/10.1007/s11134-015-9448-8
[22] N.D. Vvedenskaya, R.L. Dobrushin, and F.I. Karpelevich. 1996. Queueing System with Selection of the Shortest of Two Queues: an Asymptotic Approach. *Problemy Peredachi Informatsii* 32 (1996), 15–27.
[23] W. Wang, M. Harchol-Balter, H. Jiang, A. Scheller-Wolf, and R. Srikant. 2017. Delay Asymptotics and Bounds for Multi-Task Parallel Jobs. (2017).

A APPENDIX

A.1 Validation : Workload distribution

In this section we illustrate the accuracy of our numerical method, that is, we numerically obtain the equilibrium workload distribution for different policies from their associated FDEs and compare with results obtained via simulation. All simulation experiments are for $N = 10, 50$ and 300 servers, where we simulate the system up to time $10^7/N$ with a warm-up period of 30% and start with an empty system. The results are presented in Figure 6. The plots indicate that while the accuracy of our approximation is quite poor for $N = 10$, it becomes more and more accurate as N increases and is already very accurate for $N = 300$ in each of the considered cases. In the remainder of this section we list the parameters settings in each of the 5 examples considered in Figure 6.

In Figure 6a, we validate the Red(d, k, δ) policy found from Propositions 6.2 and 7.1 for the parameters $d = 3, k = 2, \delta = 0.02$. The slowdown S is equal to zero with probability $1 - q = 0.8$ and exponentially distributed with parameter 1 with probability $q = 0.2$. X follows a geometric distribution with parameter $1/2$ scaled down such that $\mathbb{E}[X] = 1$.

In Figure 6b, we consider the RTQ(d, T) policy with $d = 2$ and $T = 3$. We choose $\lambda = 0.75$, while the slowdown and job size distribution are chosen the same as for Red(d, k, δ) above. The equilibrium workload distribution is obtained using the combination of Propositions 6.4, 7.1 and 7.2.

In Figure 6c, we consider the DR(d, T) policy with parameters $d = 2, T = 3, \lambda = 0.7$ and S and X taken as for Red(d, k, δ) and RTQ(d, T) above. The equilibrium workload distribution is obtained using Proposition 6.6.

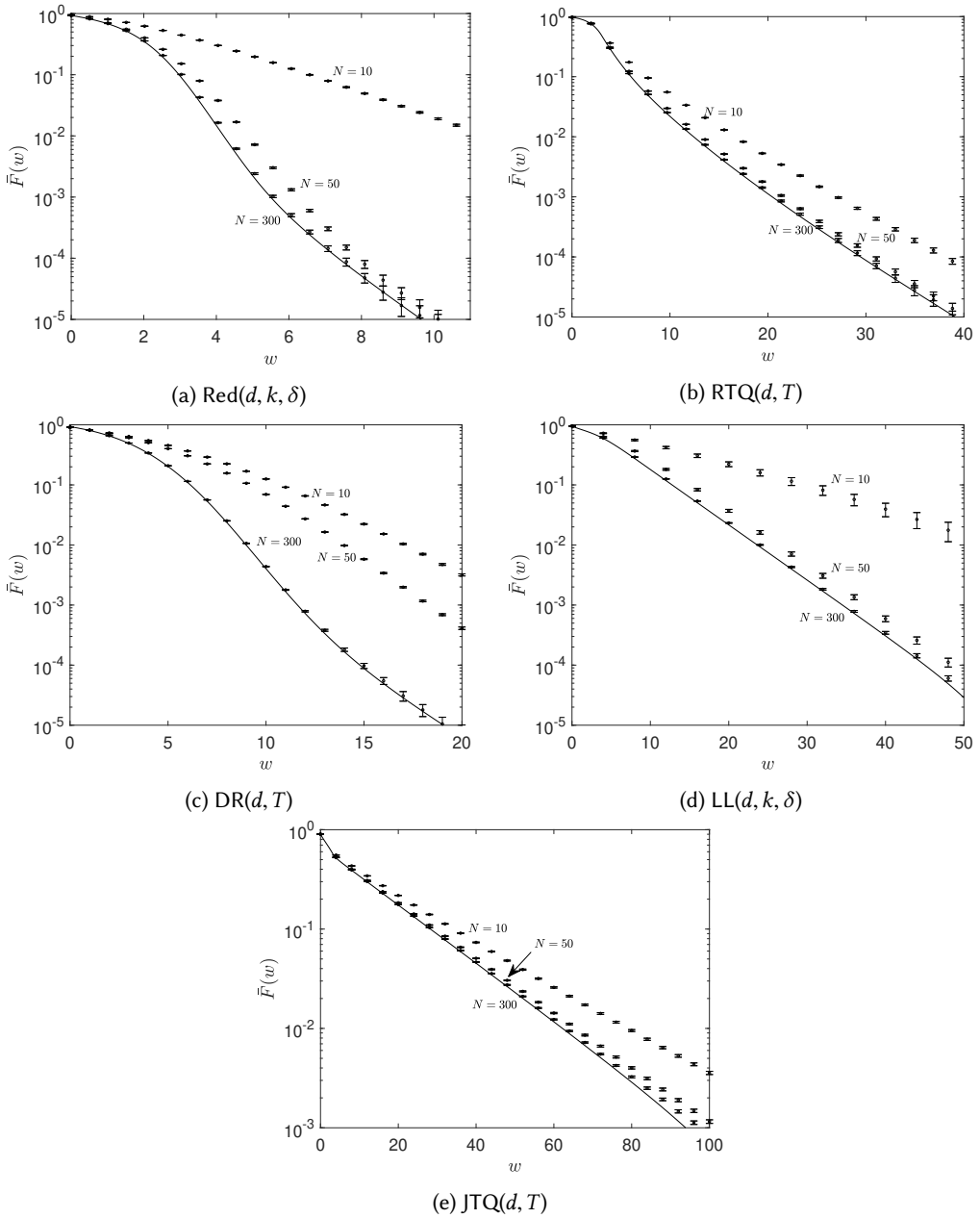
We then consider the LL(d, k, δ) policy with the following parameters. We consider $d = 3, k = 2, \delta = 0.02$ and $\lambda = 0.9$. For this policy we assume that $\tilde{X} = g(S, X)$ follows an hyperexponential distribution with two phases and balanced means (i.e. the load from the large and small jobs is the same). Furthermore, $\mathbb{E}[\tilde{X}] = 1$ and its SCV= 9. The equilibrium workload distribution is obtained using Propositions 6.8 and 7.3. The accuracy of our approximation method for large N is illustrated in Figure 6d.

Finally, in Figure 6e we consider the JTQ(d, T) policy with parameters $d = 2, T = 3, \lambda = 0.7$ with $\tilde{X}$ the same as for LL(d, k, δ). The equilibrium workload distribution is obtained using Proposition 6.11.

These plots are only a small subset of all numerical validation we have done. We have considered other values for the parameters and other slowdown/job size distributions. However the results are all similar to the ones shown in Figure 6.

A.2 Validation : Stability

In this subsection, for the DR(d, T) and the Red(d, k, δ) policy, we investigate its stability, i.e., the maximum value of arrival rate λ_{max} such that the system remains stable for $\lambda < \lambda_{\max}$, but is unstable for all $\lambda \geq \lambda_{\max}$. For the Red$(d, k, \delta)$ policy, we consider a system with $N = 300, d = 2, k = 1, \delta = 0.01$



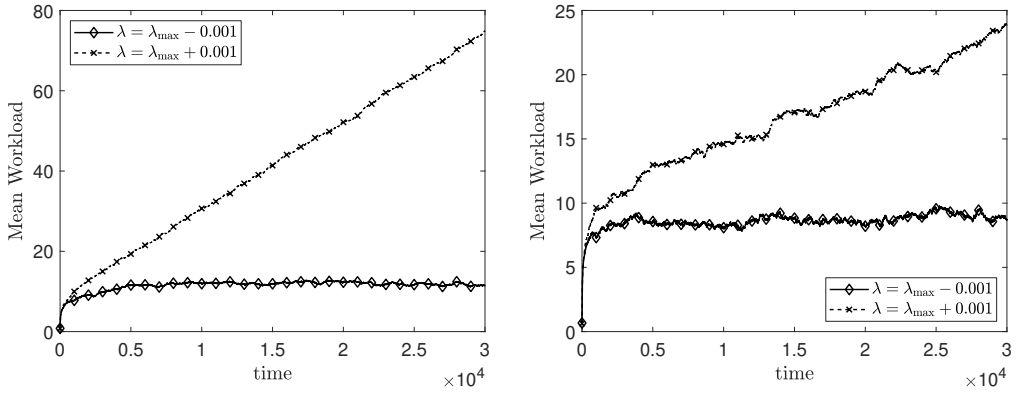
(a) Red(d, k, δ) (b) RTQ(d, T)

(c) DR(d, T) (d) LL(d, k, δ)

(e) JTQ(d, T)

Fig. 6. Limiting workload distribution vs. simulation for the N server system with Red(3, 2, 0.02), RTQ(2, 3), DR(2, 3), LL(3, 2, 0.02) and JTQ(2, 3) policies respectively under different settings of arrival rate λ, service requirement X and the slowdown variable S.

and $q = 0.2$. As earlier, we assume that S is exponential with probability q and zero with probability $1 - q$ and X is a scaled geometric random variable with parameter $1/2$ and mean 1. We obtain λ_{max} using the algorithm presented in Section 8.2. We find that for this set of parameters, we have



(a) Red(d, k, δ) (b) DR(d, T)

Fig. 7. Simulations for Red(d, k, δ) and DR(d, T) around their respective values of λ_{max}.

$\lambda_{max} = 0.7145$. To verify this, we consider a simulation of the system with $N = 300$ and values of λ just above resp. below λ_{max}. We simulate the system for a time span equal to $3 \cdot 10^4 \approx 10^7/N$ and start with an empty system. In Figure 7a, we observe that while the mean workload for $\lambda = \lambda_{max} - 0.001$ seems to converge, it appears to diverge for $\lambda = \lambda_{max} + 0.001$.

A similar experiment was performed for the DR(d, T) policy, where we consider $d = 2,, T = 3, q = 0.2$ and again the same slowdown and job size distribution. For this setting of the parameters, we observe that $\lambda_{max} = 0.7571$. Figure 7b seems to indicate that our approach to find λ_{max} is indeed quite accurate even for finite N.

A.3 Policies in practice

A.3.1 Red(d, k, δ). The Red(d, k, δ) policy is of interest in distributed storage systems with coding [12, 13, 19, 20]. Here, a file of size $k \cdot X$ is encoded into d sub-files and each of these sub-files is stored on a unique server. Due to the underlying coding scheme involved, any k of the d sub-files are sufficient to retrieve the original file. With the Red(d, k, δ) policy, once any k of the d sub-files are retrieved or downloaded, the original file can be recreated.

A.3.2 RTQ(d, T). This policy assumes the servers know their workload (or at least whether or not their workload exceeds some threshold T). This information is generally not available, therefore we could implement this policy by replicating each job onto d chosen servers and cancelling the replicas which have not yet entered service after some time T. If none of the replicas started service by time T, one replica is retained at random and the other $d - 1$ are cancelled.

Remark. *One could argue that a more realistic policy would be to assume that an incoming job is assigned to a primary server and $d - 1$ other servers are selected at random. The replicas on the other $d - 1$ non-primary servers get cancelled if they did not enter service by time T. When one of the replicas completes service, the other $d - 1$ replicas are automatically cancelled. For this policy, no information on the workload is required and the only communication needed is when one job finishes service. This policy can be studied analogously.*

A.3.3 LL(d, k, δ). The LL(d, k) policy finds its application in systems with parallel processing and multi-task jobs [23]. For such applications, one can reinterpret the LL(d, k) policy in a slightly different manner as follows. We assume that an arriving job has a service requirement of $k \cdot X$ which is split into k identical parts. An arriving job samples d servers at random and the k subtasks

of the jobs receive service from the k least-loaded servers among the d that were sampled. We say a job has been processed when all of its subtasks are processed. Note that while in $Red(d,k)$, we can have upto d subtasks being served simultaneously (due to its cancellation-on-completion nature), in LL(d,k) one can have at most k subtasks to be simultaneously in service. In this sense, LL(d,k) can be viewed, as a cancellation-on-start version of the Red(d,k) policy.

Remark. *For applications such as distributed storage, we need to consider the LL(d,k) policy with k identically distributed (not necessarily independent nor identical) jobs denoted by $X_1 \dots X_k$. Let $\tilde{Q}$ denote the Q function associated to the LL(d,k) policy with identically distributed sub-jobs. It is not hard to see that if we let $X \stackrel{d}{=} X_1$ and let $S_1, \dots, S_d$ denote independent slowdown variables we have:*

$$\tilde{Q}(U_1, \dots, U_d, S_1, \dots, S_d, X_1, \dots, X_k) = Q(U_1, \dots, U_d, S_1, \dots, S_d, X).$$

Therefore we have:

$$\mathbb{P}\{\tilde{Q}(U) > w, U \le w\} = \mathbb{P}\{Q(U) > w, U \le w\},$$

which entails that the FDE one finds for this more general model is the same as for simply taking identical job sizes. Thus, the analysis is the same as for identical job sizes, and we may restrict ourselves to the case of identically distributed sub-jobs. Note that this is indeed consistent with the encoding of X into d sub files: only the first k sub files are used.

Additional variants to LL(d,k,δ) are for example: sending the largest job to the least loaded server, the second largest to the second least loaded etc. One could also allow to send multiple jobs to the same server if that server has a very low load.

A.4 Proof of Theorem 5.1

Proof. Let $\Delta > 0$ be arbitrary, and consider the events where the workload of the queue at cavity becomes w at time $t+\Delta$ by looking at its value at time t and the behaviour in $[t, t+\Delta]$. As potential arrivals occur according to a Poisson process, all events which involve more than one arrival in $[t, t+\Delta]$ are $o(\Delta)$. We do distinguish the following three events which do not involve more than one arrival on $[t, t+\Delta]$:

- The queue at the cavity has workload $w+\Delta$ at time t and its workload is not increased by arrivals in $[t, t+\Delta]$, this occurs with density:

$$Q_1 = \left(1 - \lambda d \int_0^{\Delta} \mathbb{P}\left\{Q(U^{\mathcal{H}(\cdot)}((t+v)-) > w+\Delta-v, U^{\mathcal{H}(\cdot)}((t+v)-) = w+\Delta-v\right\} dv\right).$$

- The queue at the cavity has workload 0 at time t and its workload is increased to $w+(\Delta-v)$ at time $t+v$. This event has density:

$$Q_2 = \lambda d \int_0^{\Delta} \mathbb{P}\left\{Q(U^{\mathcal{H}(\cdot)}((t+v)-) = w+(\Delta-v), U^{\mathcal{H}(\cdot)}((t+v)-) = 0\right\} dv.$$

- The queue at the cavity's workload lies in the interval $(v, w+\Delta-v)$ at time $t+v$ and its workload is increased to $w+\Delta-v$ by a potential arrival, we find:

$$Q_3 = \lambda d \int_0^{\Delta} \int_v^{w+\Delta-v} \mathbb{P}\{Q(U^{\mathcal{H}(\cdot)}((t+v)-) = w+(\Delta-v), U^{\mathcal{H}(\cdot)}((t+v)-) = u\}\, du\, dv.$$

We therefore find that:

$$f(t+\Delta, w) = Q_1 + Q_2 + Q_3 + o(\Delta),$$

by subtracting $f(t,w)$ on both sides, dividing by Δ and taking the limit $\Delta \to 0$ we find that the claimed equality (1) indeed holds. We now investigate the boundary condition which characterizes the events associated with the evolution of the workload on $[t, t+\Delta]$ such that it is zero at time

$t+\Delta$. The workload can either be zero at time t and no potential arrivals occur in $[t, t+\Delta]$ or the workload is in $(0,\Delta)$ at time t and no potential arrival in the interval $[t, t+\Delta]$ increases its workload. We find:

$$F(t+\Delta,0) = F(t,0)\left(1-\lambda d\int_0^{\Delta}\mathbb{P}\{Q(U^{\mathcal{H}(\cdot)}((t+v)-)>0 \mid U^{\mathcal{H}(\cdot)}(t)=0\})\,dv\right)$$
$$+\int_0^{\Delta} f(t,u)\left(1-\lambda d\int_0^{\Delta}\mathbb{P}\{Q(U^{\mathcal{H}(\cdot)}((t+v)-))>\max\{0,(u-v)\} \mid U^{\mathcal{H}(\cdot)}(t)=u\}\right)dv\,du+o(\Delta).$$

By subtracting $F(t,0)$ on both sides, dividing by Δ and taking the limit $\Delta\to 0$ we find that (2) indeed holds. □

A.5 Proof of Proposition 6.2

Proof. Assume that the selected servers have workloads $U_1,\dots,U_d$ and an arriving job of size X experiences slowdown $S_1,\dots,S_d$ on the selected servers. If $w\le\delta$, it is obvious that $Q(U_1)>w$ if and only if its response time at the queue at the cavity $R_{\tilde{X}}\overset{d}{=}U_1+g(S_1,X)>w$. We therefore find from Theorem 5.2 for $w\le\delta$:

$$\bar{F}'(w) = -\lambda d\mathbb{P}\{U_1+g(S_1,X)>w, U_1\le w\},$$

which leads to (8). Now, assume $w>\delta$. For $i=1,\dots,d$ we define $Y_i=U_i+g(S_i,X)$ as the *effective workload* after addition of a copy. We have

$$Q(U_1) = \max\left\{U_1, \min\{Y_1, Y_{(k)}+\delta\}\right\}. \tag{21}$$

In words, this means that the new workload at the cavity queue can take one of the values below due to the potential arrival.

- $Q(U_1)=U_1$. This happens when $U_1>Y_{(k)}+\delta$.
- $Q(U_1)=Y_1$. This happens when $Y_1<Y_{(k)}+\delta$.
- $Q(U_1)=Y_{(k)}+\delta$. This happens when $Y_1\ge Y_{(k)}+\delta$ and $U_1\le Y_{(k)}+\delta$.

Let us consider the term $\mathbb{P}\{U_1\le w, Q(U_1)>w\}$. Note that the event $\{U_1\le w, Q(U_1)>w\}$ implies that $Y_1>w$. Further, $Y_1>w$ implies that $Q(U_1)\ne U_1$ and hence from (21) we have:

$$\mathbb{P}\{U_1\le w, Q(U_1)>w\} = \mathbb{P}\{U_1\le w, \min\{Y_1,Y_{(k)}+\delta\}>w\} = \mathbb{P}\{U_1\le w, Y_1>w, Y_{(k)}+\delta>w\}.$$

Now $Y_{(k)}+\delta>w$ only if at most $k-1$ of the sampled queues have an *effective workload* which is bounded above by $w-\delta$. Given that $Y_1>w$ for the queue at cavity, we have that its effective workload is not bounded by $w-\delta$. By conditioning on the random variable X, we have from Theorem 5.2, the ansatz property (that the queues have independent workloads) and the discussion above that (9) indeed holds. □

A.6 Proof of Proposition 6.4

Proof. We again assume a potential arrival of size X occurs to servers with workloads $U_1,\dots,U_d$ where it experiences slowdown $S_1,\dots,S_d$. It is not hard to see that for the RTQ(d,T) policy, we have:

$$Q(U_1) = \begin{cases} \max\left\{U_1, \min_{i=1}^{d}\{U_i+g(S_i,X) \mid U_i\le T\}\right\} & \text{if } U_1\le T \\ U_1+g(S_1,X) \text{ w.p. } \frac{1}{d} \quad \text{if } U_1,\dots,U_d>T & \\ U_1 & \text{otherwise,}\end{cases}$$

where $\min_{i=1}^{d}\{U_i+g(S_i,X)\mid U_i\le T\}$ is the minimum taken over those i for which $U_i\le T$. We compute the probability $\mathbb{P}\{Q(U_1)>w, U_1\le w, X=x\}$ the result then follows from Theorem 5.2

(after integrating out X). For $w \leq T$ we find that $\mathbb{P}\{\boldsymbol{Q}(U_1) > w, U_1 \leq w, X = x\}$ is equal to:

$$\mathbb{P}\{U_1 + g_x(S_1) > w, U_1 \leq w\} \cdot \mathbb{P}\{(U_1 + g_x(S)) > w \text{ or } U_1 > T\}^{d-1}, \tag{22}$$

but note that if $U > T$ then surely also $U + g_x(S) > w$ holds. Therefore, (22) further simplifies to:

$$\bar{F}_{R_x}(w)^{d-1}(\bar{F}_{R_x}(w) - \bar{F}(w)),$$

this shows the first part. Assume $w > T$, we split $\mathbb{P}\{\boldsymbol{Q}(U_1) > w, U_1 \leq w, X = x\}$ by writing it as $\mathbb{P}\{\boldsymbol{Q}(U_1) > w, U_1 \leq T, X = x\} + \mathbb{P}\{\boldsymbol{Q}(U_1) > w, T < U_1 \leq w, X = x\}$. When $T < U_1$, the workload can only increase when the workload at all other servers also exceed T, therefore we have

$$\mathbb{P}\{\boldsymbol{Q}(U_1) > w, T < U_1 \leq w, X = x\} = \frac{1}{d}\bar{F}(T)^{d-1}\mathbb{P}\{U_1 + g_x(S) > w, T < U_1 \leq w\}. \tag{23}$$

For the workload to jump from below T to above w, one requires that all the other selected queues either have a workload which exceeds T (thus the job will not be replicated on them) or a response time which exceeds w. This allows us to find:

$$\mathbb{P}\{\boldsymbol{Q}(U_1) > w, U_1 \leq T, X = x\} = \mathbb{P}\{U_1 + g_x(S) > w, U_1 \leq T\} \cdot (\bar{F}(T) + \mathbb{P}\{U_1 + g_x(S) > w, U_1 \leq T\})^{d-1} \tag{24}$$

It is not hard to verify that $B_x(w, T) = \mathbb{P}\{U_1 + g_x(S) > w, U_1 \leq T\}$. Combining (22-24) with this equality, we may conclude the proof. □

A.7 Proof of Corollary 6.5

Proof. For $w \leq T$ we find that all selected queues which have a workload that exceeds T won't be able to finish the job in time w. For each of the queues which have a workload which does not exceed T, we find that $B_x(w, T)$ is the probability that they won't finish the job before time w. For $w > T$ we first have a term corresponding to the case where all d selected queues have a workload which exceeds T. In the second term we consider the case where k out of d selected queues have a workload smaller than T. □

A.8 Proof of Proposition 6.6

Proof. In order to apply Theorem 5.2, we assume a potential arrival occurs to queues with workloads $U_1, \ldots, U_d$, and where jobs with job size equal to X experience a slowdown equal to $S_1, \ldots, S_d$. We make the distinction whether or not the queue at the cavity is the primary server which is initially selected. We find that $\mathbb{P}\{\boldsymbol{Q}(U_1) > w, U_1 \leq w\}$ equals:

$$\mathbb{P}\{\boldsymbol{Q}(U_1) > w, U_1 \leq w, U_1 \text{ not the primary server }\} \tag{25}$$

$$+ \mathbb{P}\{\boldsymbol{Q}(U_1) > w, U_1 \leq w, U_1 \text{ the primary server }\}. \tag{26}$$

For (25) we obtain that it is equal to:

$$\frac{d-1}{d}\int_0^\infty (\bar{F}_{R_x}(w) - \bar{F}(w))\bar{F}_{R_x}(w)^{d-2} \cdot \bar{F}_{R_x}(w+T) f_X(x)\, dx, \tag{27}$$

which reads: the queue at the cavity has a response time which exceeds w but its workload does not exceed w, the other $d-2$ selected servers' response times all exceed w and the primary server's response time exceeds $w + T$. For (26) we make a distinction between the cases $w \leq T$ and $w > T$. For $w \leq T$, we find that it equals:

$$\frac{1}{d}\int_0^\infty (\bar{F}_{R_x}(w) - \bar{F}(w)) f_X(x)\, dx, \tag{28}$$

while for $w > T$, we find that the job is replicated onto the other servers after time T thus to finish after the queue at the cavity reaches workload w, they must process the job in at least $w - T$ time.

$$\frac{1}{d}\int_0^\infty (\bar{F}_{R_x}(w) - \bar{F}(w)) \cdot \bar{F}_{R_x}(w-T)^{d-1} f_X(x)\, dx, \tag{29}$$

where we note that $\bar{F}_{R_x}(w) = 1$ if $w < 0$. Putting (27-29) together, we obtain the sought result. □

A.9 Proof of Proposition 6.8

Proof. Suppose that at some arbitrary point in time at equilibrium, a potential arrival of size $k \cdot X$ arrives to servers with queue lengths $U_1, \ldots, U_d$ and slowdowns $S_1, \ldots, S_d$. From Theorem 5.2 we find that the equilibrium workload environment satisfies:

$$\bar{F}'(w) = -\lambda d(\mathbb{P}\{\boldsymbol{Q}(U) > w, U \le w\} = -\lambda d\mathbb{P}\{\boldsymbol{Q}(U) > w, U = 0\} \tag{30}$$

$$- \lambda d\mathbb{P}\{\boldsymbol{Q}(U) > w, 0 < U \le w\}. \tag{31}$$

It can be seen that (30) is equal to the following:

$$-\lambda d F(0)\bar{F}_{\tilde{X}}(w) \cdot \sum_{j=0}^{d-1} \min\left\{1, \frac{k}{j+1}\right\} \binom{d-1}{j} \bar{F}(0)^{d-1-j} F(0)^j = -\lambda \bar{F}_{\tilde{X}}(w)(H(0)+1) \tag{32}$$

Similarly, (31) is equal to the following:

$$-\lambda d \int_0^w f(u) \sum_{j=0}^{k-1} \binom{d-1}{j} (1-\bar{F}(u))^j \bar{F}(u)^{d-j-1} \bar{F}_{\tilde{X}}(w-u)\, du. \tag{33}$$

To further simplify this, we first define

$$h(u) = f(u) \sum_{j=0}^{k-1} \binom{d-1}{j} (1-\bar{F}(u))^j \bar{F}(u)^{d-j-1}$$

and show that $H'(w) = d \cdot h(w)$ where $H(w)$ is given by (17). Towards this, note that

$$H'(w)/f(w) = -\sum_{j=1}^{k} j(d-j)\binom{d}{j} \bar{F}(w)^{d-j-1}(1-\bar{F}(w))^j \tag{34}$$

$$- \sum_{j=k+1}^{d} k(d-j)\binom{d}{j} \bar{F}(w)^{d-j-1}(1-\bar{F}(w))^j \tag{35}$$

$$+ \sum_{j=1}^{k} j^2 \binom{d}{j} \bar{F}(w)^{d-j}(1-\bar{F}(w))^{j-1} \tag{36}$$

$$+ \sum_{j=k+1}^{d} kj \binom{d}{j} \bar{F}(w)^{d-j}(1-\bar{F}(w))^{j-1}. \tag{37}$$

It is not hard to see that (36) is equal to:

$$\sum_{j=0}^{k-1} (j+1)\binom{d}{j}(d-j)\bar{F}(w)^{d-j-1}(1-\bar{F}(w))^j.$$

This allows one to find that the sum of (34) and (36) simplifies to the following.

$$\sum_{j=0}^{k-1}(d-j)\binom{d}{j}\bar{F}(w)^{d-j-1}(1-\bar{F}(w))^j - k(d-k)\binom{d}{k}\bar{F}(w)^{d-k-1}(1-\bar{F}(w))^k. \tag{38}$$

Similarly (37) can be re-written as

$$k\sum_{j=k}^{d-1}(d-j)\binom{d}{j}\bar{F}(w)^{d-j-1}(1-\bar{F}(w))^j$$

and adding this to (35) gives us the following.

$$k(d-k)\binom{d}{k}\bar{F}(w)^{d-k-1}(1-\bar{F}(w))^k. \tag{39}$$

The addition of (38) and (39) equals $d \cdot h(w)$ which proves that $H'(w) = d \cdot h(w)$. We split (33) over its continuous and discrete part using $\bar{F}_{\tilde{X}} = \bar{F}_{\tilde{X}_c} + \bar{F}_{\tilde{X}_d}$. Using integration by parts and the fact that $H'(u) = d \cdot h(u)$, for the continuous part we find:

$$\int_0^w d \cdot h(u)\bar{F}_{\tilde{X}_c}(w-u)\,du = \bar{F}_{\tilde{X}_c}(0)H(w) - H(0)\bar{F}_{\tilde{X}_c}(w) - \int_0^w H(u)f_{\tilde{X}_c}(w-u)\,du. \tag{40}$$

For the discrete part we find by setting $\iota(w) = \max\{n \mid x_n \le w\}$ that:

$$\int_0^w d \cdot h(u)F_{\tilde{X}_d}(w-u)\,du = \bar{F}_{\tilde{X}_d}(0)H(w) - H(0)\bar{F}_{\tilde{X}_d}(w) - \sum_{j=0}^{\iota(w)} p_j H(w-x_j). \tag{41}$$

Using (32) for (30) and the combination of (40-41) for (31) we find that the sought equality indeed holds.

The FPE (18) follows by integrating (16) w.r.t. w and applying Fubini. □

A.10 Proof of Proposition 6.9

Proof. To show the first part, let $\bar{F} \in \text{ccdf}_\rho$ be arbitrary. We note that:

$$1 + H(w) \le k \cdot \sum_{j=0}^{d}\binom{d}{j}\bar{F}(w)^{d-j}(1-\bar{F}(w))^j \le k,$$

which shows that $\lim_{w\to\infty} T_d^{(k)}\bar{F}(w) \le 0$. To show the other inequality, we note that for large values of w we have $1 + H(w) \ge k \cdot (1-\bar{F}(w))^j$.

To show (20) we let $\bar{F}_1, \bar{F}_2 \in \text{ccdf}_\rho$. It is clear that we should bound:

$$\left|\sum_{j=1}^{d}\min\{j,k\}\binom{d}{j}\left(\bar{F}_1(w)^{d-j}(1-\bar{F}_1(w))^j - \bar{F}_2(w)^{d-j}(1-\bar{F}_2(w))^j\right)\right|.$$

To this end, we define the function $f_{j,d}(x) = \sum_{j=1}^{d} \min\{j,k\}\binom{d}{j}x^j \cdot (1-x)^{d-j}$. We may bound its first derivative by:

$$\begin{aligned} f'_{j,d}(x) &= \sum_{j=1}^{d} \min\{j,k\}\binom{d}{j}(1-x)^{d-j-1}x^{j-1}(j-d\cdot x) \\ &\leq k\sum_{j=1}^{d} j\binom{d}{j}(1-x)^{d-j-1}x^{j-1} - d\sum_{j=1}^{d}\min\{j,k\}\binom{d}{j}(1-x)^{d-j-1}x^{j} \\ &= d\sum_{j=1}^{k-1}(k-j)(1-x)^{d-j-1}x^{j}. \end{aligned}$$

By applying the mean value theorem, this completes the proof. □

A.11 Response time for the LL(d, k, δ) policy

The queue with the highest workload need not be the last queue to finish serving its part of the job. The response time of a job is given by $R = \max_{i=1}^{k}\{U_{(i)} + g(S_i, X)\}$. We find:

$$\bar{F}_R(w+\delta) = 1 - \int_0^\infty \mathbb{P}\left\{\max_{i=1}^{k}\{U_{(i)} + g_x(S_i)\} \leq w\right\} f_X(x)\,dx$$

and $\bar{F}_R(w) = 1$ for $w \leq \delta$. By applying the fact that the pdf of the joint distribution $(U_{(1)}, \ldots, U_{(k)})$ in $u_1, \ldots, u_k$ is given by $\frac{d!}{(d-k)!}f(u_1)\cdot\cdots\cdot f(u_k)\bar{F}(u_k)^{n-k}$ we find:

$$\begin{aligned} \mathbb{P}\{\max_{i=1}^{k}\{U_{(i)} + g_{\frac{x}{k}}(S_i) \leq w\}\} = \frac{d!}{(d-k)!}\int_0^w f(u_1)F_{g_{\frac{x}{k}}(S)}(w-u_1)\int_{u_1}^w f(u_2)F_{g_{\frac{x}{k}}(S)}(w-u_2) \\ \int_{u_2}^w \cdots \int_{u_{k-1}}^w F_{g_{\frac{x}{k}}(S)}(w-u_k)f(u_k)\bar{F}(u_k)^{d-k}\,du_k\ldots du_1. \end{aligned} \tag{42}$$

When $d > k \geq 2$, this integral becomes hard to solve, we may therefore first compute the workload distribution and thereafter use simulation to obtain the response time distribution. More precisely, whenever an arrival occurs, we simulate $X, S_1, \ldots, S_d$ according to their distribution and $U_1, \ldots, U_d$ as i.i.d. random variables distributed as the obtained workload distribution. One can then simply apply the LL(d, k, δ) policy to obtain the response time for this specific set of simulated values. Note that in this simulation one need not keep track of any values, simply simulate arrivals and compute their response times based on the obtained workload distribution.

A.12 Proof of Proposition 6.11

Proof. Note that for this policy, $Q(U_1) = Q(U_1, \ldots, U_d, \tilde{X})$ is given by:

$$\begin{aligned} Q(U_1) &= U_1 + \tilde{X} && w.p.\ \frac{1}{|\{k \in \{1,\ldots,d\} \mid U_k \leq T\}|} \text{ if } U_1 \leq T \\ Q(U_1) &= U_1 + \tilde{X} && w.p.\ \frac{1}{d} \text{ if } U_1, \ldots, U_d > T \\ Q(U_1) &= U_1 && \text{otherwise.} \end{aligned}$$

We first compute the probability $\mathbb{P}\{Q(U_1) > w, U_1 \leq w\}$ for the case $w \leq T$. We find that it is equal to:

$$\left(\sum_{j=0}^{d-1} \binom{d-1}{j} F(T)^j \bar{F}(T)^{d-1-j} \frac{1}{j+1}\right) \mathbb{P}\{U_1 + \tilde{X} > w, U_1 \leq w\}$$
$$= \left(\frac{1}{d} \sum_{j=1}^{d} \binom{d}{j} \frac{F(T)^j \bar{F}(T)^{d-j}}{F(T)}\right) \mathbb{P}\{U_1 + \tilde{X} > w, U_1 \leq w\} = \frac{1 - \bar{F}(T)^d}{dF(T)} \mathbb{P}\{U_1 + \tilde{X} > w, U_1 \leq w\}.$$

This shows the first part. Assume $w > T$, we first write $\mathbb{P}\{Q(U_1) > w, U_1 \leq w\}$ as $\mathbb{P}\{Q(U_1) > w, U_1 \leq T\} + \mathbb{P}\{Q(U_1) > w, T < U_1 \leq w\}$. We then find that $\mathbb{P}\{Q(U_1) > w, U_1 \leq T\}$ is given by:

$$\frac{(1 - \bar{F}(T)^d)}{dF(T)} \mathbb{P}\{U_1 + \tilde{X} > w, U_1 \leq T\}.$$

while $\mathbb{P}\{Q(U_1) > w, T < U_1 \leq w\}$ is given by:

$$\frac{1}{d}\mathbb{P}\{U_2, \ldots, U_d > T, U + \tilde{X} > w, T < U \leq w\} = \frac{1}{d}\bar{F}(T)^{d-1}(\bar{F}_{R_X}(w) - \bar{F}(w) - \mathbb{P}\{U \leq T, U + \tilde{X} > w\}).$$

One finds that $A(w, T) = \mathbb{P}\{U \leq T, U + \tilde{X} > w\}$, which completes the proof. □

A.13 No Slowdown

In this subsection we take another look at some of the policies studied in Section 6, and revisit them under the assumption that the servers experience no slowdown (i.e. $g(S, X) = X$). We first note that the analysis for LL(d, k, δ) and JTQ(d, T) under the no slowdown assumption easily follows by taking $\tilde{X} = X$. We now focus on the Red(d, k, δ), RTQ(d, T) and DR(d, T) policy for the case without slowdown.

A.13.1 Red(d, k, δ). For Red(d, k, δ) with no slowdown, we find that the results in Proposition 6.2 can be simplified, which allows to obtain an analog to Proposition 7.1 in case X is PH distributed.

Proposition A.1. *The ccdf of the workload distribution for Red(d, k, δ) without slowdown satisfies the following DIDE:*

$$\bar{F}'(w) = -\lambda d\left(\bar{F}_X(w) + \int_0^w f_X(w-u)\bar{F}(u)\,du - \bar{F}(w)\right) \qquad w \leq \delta \tag{43}$$

$$\bar{F}'(w) = -\lambda d\left[\bar{F}_X(w) - \bar{F}(w)\bar{F}_X(w-\delta) + \int_0^{\delta} \bar{F}(x) f_X(w-x)\,dx + \int_0^{w-\delta} L(w-x-\delta)(\bar{F}(w-x) - \bar{F}(w)) f_X(x)\,dx\right] \qquad w > \delta \tag{44}$$

with $L(w) = \sum_{j=0}^{k-1} \binom{d-1}{j} F(w)^j \bar{F}(w)^{d-j-1}$. *Furthermore if* X *is PH distributed with parameters* (α, A) *and* $\mu = -A\mathbf{1}$ *we find that the above DIDE reduces to a DDE:*

$$\begin{aligned}
\bar{F}'(w) &= -\lambda d \left(\bar{F}_X(w) + \alpha\xi_1(w) - \bar{F}(w)\right) && w \le \delta \\
\bar{F}'(w) &= -\lambda d \left(\bar{F}_X(w) - \bar{F}(w)\bar{F}_X(w-\delta) + \alpha\left(\xi_1(w) + \xi_2(w) - \xi_3(w)\bar{F}(w)\right)\right) && w > \delta \\
\xi_1'(w) &= A\xi_1(w) + \bar{F}(w)\mu && w \le \delta \\
\xi_1'(w) &= A\xi_1(w) && w > \delta \\
\xi_2'(w) &= L(w-\delta)\bar{F}(w)\mu + A\xi_2(w) && w > \delta \\
\xi_3'(w) &= L(w-\delta)\mu + A\xi_3(w) && w > \delta.
\end{aligned}$$

with boundary condition $\xi_1(0) = \xi_2(\delta) = \xi_3(\delta) = 0$.

Proof. We find:

$$\bar{F}_{R_{\tilde{X}}}(w) = \mathbb{P}\{U + X \ge w\} = \bar{F}_X(w) + \int_0^w f_X(w-x)\bar{F}(x)\,dx.$$

Moreover we have:

$$\bar{F}_{R_x}(w) = \bar{F}(w-x).$$

This allows us to compute:

$$\begin{aligned}
\bar{F}'(w) &= -\lambda d \int_0^\infty L(w-x-\delta)\left(\bar{F}(w-x) - \bar{F}(w)\right) f_X(x)\,dx \\
&= -\lambda d \int_0^w L(w-x-\delta)(\bar{F}(w-x) - \bar{F}(w)) f_X(x)\,dx - \lambda d \bar{F}_X(w)F(w).
\end{aligned}$$

From this it is clear that (43-44) holds.

Now assume that X is PH distributed with parameters (α, A). This last statement follows by defining:

$$\begin{aligned}
\xi_1(w) &= \int_0^w e^{(w-x)A}\bar{F}(x)\,dx\mu && w \le \delta \\
\xi_1(w) &= \int_0^\delta e^{(w-x)A}\bar{F}(x)\,dx\mu && w > \delta \\
\xi_2(w) &= \int_0^{w-\delta} L(x)\bar{F}(x+\delta)e^{(w-x-\delta)A}\mu\,dx \\
\xi_3(w) &= \int_0^{w-\delta} L(x)e^{(w-x-\delta)A}\mu\,dx
\end{aligned}$$

□

A.13.2 RTQ(d). For RTQ(d, T) with no slowdown, we find that the result in Proposition 6.4 can be simplified, which allows to obtain an analogous simplification to Proposition 7.1, 7.2 in case X is PH distributed.

Proposition A.2. *The ccdf of the workload distribution for RTQ(d) without slowdown satisfies the following DIDE:*

$$\bar{F}'(w) = -\lambda d\left[F(w)\bar{F}_X(w) + \int_0^w \bar{F}(w-x)^{d-1}\left(\bar{F}(w-x) - \bar{F}(w)\right) f_X(x)\,dx\right] \qquad w \le T \quad (45)$$

$$\begin{aligned}\bar{F}'(w) = &-\lambda d\left[F(T)\bar{F}_X(w) + \int_0^T \bar{F}(T-x)^{d-1}\left(\bar{F}(T-x) - \bar{F}(T)\right) f_X(x+w-T)\,dx\right] \\ &- \lambda\bar{F}(T)^{d-1}\left[\left(\bar{F}(T) - \bar{F}(w)\right)\bar{F}_X(w-T) + \int_0^{w-T}\left(\bar{F}(w-x) - \bar{F}(w)\right) f_X(x)\,dx\right] \qquad w > T \quad (46)\end{aligned}$$

When X has a PH distribution with parameters (α, A) and $\mu = -A\mathbf{1}$ we find that (45-46) simplifies to the following DDE:

$$\begin{aligned}
\bar{F}'(w) &= -\lambda d\left[F(T)\bar{F}_X(w) + \alpha\cdot\left(\xi_1(w) - \xi_2(w)\bar{F}(w)\right)\right] && w \le T\\
\bar{F}'(w) &= -\lambda d\left[F(T)\bar{F}_X(w) + \alpha\left(\xi_1(w) - \xi_2(w)\bar{F}(T)\right)\right] \\
&\quad - \lambda\bar{F}(T)^{d-1}\left[(\bar{F}(T) - \bar{F}(w))\bar{F}_X(w-T) + \alpha\left(\xi_3(w) - \bar{F}(w)F_X(w-T)\right)\right] && w > T\\
\xi_1'(w) &= A\xi_1(w) + \bar{F}(w)^d\mu && w \le T\\
\xi_1'(w) &= A\xi_1(w) && w > T\\
\xi_2'(w) &= A\xi_2(w) + \bar{F}(w)^{d-1}\mu && w \le T\\
\xi_2'(w) &= A\xi_2(w) && w > T\\
\xi_3'(w) &= A\xi_3(w) + \bar{F}(w)\mu.
\end{aligned}$$

With boundary condition $\xi_1(0) = \xi_2(0) = \xi_3(T) = 0$.

Proof. We first note that we have $\bar{F}_{R_x}(w) = \bar{F}(w-x)$ and $B_x(w,T) = 0$ if $T \le w - x$ and $\bar{F}(w-x) - \bar{F}(T)$ if $w - x < T$. This allows us to find for $w \le T$:

$$\bar{F}'(w) = -\lambda d\int_0^\infty f_X(x)\bar{F}(w-x)^{d-1}\left(\bar{F}(w-x) - \bar{F}(w)\right)\,dx$$

which easily simplifies to (45). For $w > T$ we obtain:

$$\begin{aligned}\bar{F}'(w) = &-\lambda d\int_{w-T}^\infty \left(\bar{F}(w-x) - \bar{F}(T)\right)\bar{F}(w-x)^{d-1} f_X(x)\,dx \\ &- \lambda\int_0^\infty\left[\int_0^{w-T}\left(\bar{F}(w-x) - \bar{F}(w)\right) f_X(x)\,dx + \int_{w-T}^\infty\left(\bar{F}(T) - \bar{F}(w)\right) f_X(x)\,dx\right].\end{aligned} \quad (47)$$

In order to conclude that (46) indeed holds, it suffices to note that (47) is equal to:

$$-\lambda d\left[\int_{w-T}^w \left(\bar{F}(w-x) - \bar{F}(T)\right)\bar{F}(w-x)^{d-1} f_X(x)\,dx + \int_w^\infty (1 - \bar{F}(T)) f_X(x)\,dx\right].$$

The result for PH distributed job sizes X follows by defining:

$$\xi_1(w) = \int_0^w \bar{F}(w-x)^d e^{xA}\mu dx \qquad w \le T$$

$$\xi_1(w) = \int_0^T \bar{F}(T-x)^d e^{(x+w-T)A}\, dx \qquad w > T$$

$$\xi_2(w) = \int_0^w \bar{F}(w-x)^{d-1} e^{xA}\mu\, dx \qquad w \le T$$

$$\xi_2(w) = \int_0^T \bar{F}(T-x)^{d-1} e^{(x+w-T)A}\, dx \qquad w > T$$

$$\xi_3(w) = \int_0^{w-T} \bar{F}(w-x) e^{xA}\mu\, dx.$$

□

A.13.3 DR(d, T).

Proposition A.3. *The ccdf of the workload distribution for DR(d, T) without slowdown satisfies the following FDE:*

$$\bar{F}'(w) = -\lambda\Bigg[\int_0^w f_X(x)(\bar{F}(w-x)-\bar{F}(w))((d-1)(\bar{F}(w-x)^{d-2}\bar{F}(w+T-x)+1))\,dx$$
$$+(1-\bar{F}(w))\int_0^T f_X(w+x)((d-1)\bar{F}(T-x)+1)\,dx + d\bar{F}_X(w+T)(1-\bar{F}(w))\Bigg] \qquad w \le T$$

$$\bar{F}'(w) = -\lambda\Bigg[\int_0^{w-T} (\bar{F}(w-x)-\bar{F}(w))((d-1)(\bar{F}(w-x)^{d-2}\bar{F}(w+T-x)+\bar{F}(w-T-x)^{d-1})$$
$$f_X(x)\,dx + \int_{w-T}^w f_X(x)(\bar{F}(w-x)-\bar{F}(w))((d-1)\bar{F}(w-x)^{d-2}\bar{F}(w+T-x)+1)\,dx$$
$$+(1-\bar{F}(w))\int_0^T f_X(x+w)((d-1)\bar{F}(w-x)+1)\,dx + (1-\bar{F}(w))\bar{F}_X(w+T)\Bigg] \qquad w > T.$$

Proof. This follows from Proposition 6.6 by simple computation. □

Received February 2019; revised March 2019; accepted April 2019