Performance of Redundancy(d) with Identical/Independent Replicas

TIM HELLEMANS AND BENNY VAN HOUDT, University of Antwerp

Queueing systems with redundancy have received considerable attention recently. The idea of redundancy is to reduce latency by replicating each incoming job a number of times and to assign these replicas to a set of randomly selected servers. As soon as one replica completes service the remaining replicas are cancelled. Most prior work on queueing systems with redundancy assumes that the job durations of the different replicas are i.i.d., which yields insights that can be misleading for computer system design.

In this paper we develop a differential equation, using the cavity method, to assess the workload and response time distribution in a large homogeneous system with redundancy without the need to rely on this independence assumption. More specifically, we assume that the duration of each replica of a single job is identical across the servers and follows a general service time distribution.

Simulation results suggest that the differential equation yields exact results as the system size tends to infinity and can be used to study the stability of the system. We also compare our system to the one with i.i.d. replicas and show the similarity in the analysis used for independent resp. identical replicas.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Redundancy(d), Large Scale Computer Network, Load Balancing

ACM Reference Format:
Tim Hellemans and Benny Van Houdt. 2018. Performance of Redundancy(d) with Identical/Independent Replicas. *ACM Trans. Web* ?, ?, Article ? (May 2018), 28 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Redundancy is regarded as an effective technique to reduce latency in a variety of systems including large scale computer clusters [1, 8, 9, 11]. The idea of redundancy is to create a number of replicas of each incoming job and to assign these replicas to a set of random servers. When the first of these replicas is processed by a server, the remaining replicas get cancelled. An attractive feature of this scheme is that the replicas can be assigned immediately without the need to consult the server states or the need to maintain such information. Queueing models to study the effect of redundancy on the job response time have been introduced recently (e.g., [2, 6]). One of the key assumptions to enable their analysis often exists in assuming that the processing times of the replicas are independent and identically distributed (i.i.d.) across servers. While this may be applicable in some contexts, this assumption may result in misleading insights in a computer systems setting. For instance this i.i.d. assumption suggests that mean response time reduces as a function of the number of replicas (for sufficiently variable job sizes), while without such an assumption the mean response time may increase sharply if too many replicas are used.

Author's address: Tim Hellemans and Benny Van Houdt, University of Antwerp, Middelheimlaan 1, B2020 Antwerp, Belgium, tim.hellemans@uantwerpen.be, benny.vanhoudt@uantwerpen.be.

1559-1131/2018/5-ART? $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

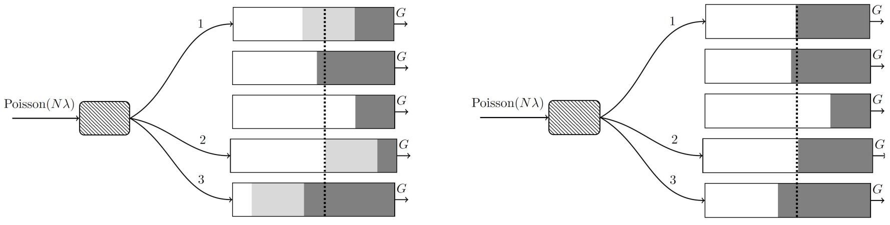
In this paper we present a fixed point equation, based on the cavity process, to assess the workload and response time distribution of a queueing model with redundancy when the processing times of the replicas are assumed to be *identical* across servers as opposed to assuming they are i.i.d.. Next, we rewrite this fixed point equation as a Delayed Integro-Differential Equation (DIDE) for general job sizes which have no atom in zero, which reduces to a Delayed Differential Equation (DDE) in case the job sizes are discrete and an Integro-Differential Equation (IDE) in case job sizes are continuous. For phase type distributed (PH-distributed) job sizes, we are able to simplify this IDE further to an ODE. We conjecture that (when the queueing system is stable) this DIDE has a unique solution that corresponds to the limit of the workload distribution as the number of servers tends to infinity. We propose a numerical scheme to solve the DIDE and illustrate that its accuracy improves with the system size for various job size distributions (i.e., for bounded Pareto, (hyper)exponential and deterministic job sizes) using simulation. We also show how this DIDE leads to a method to accurately obtain the stability region for a given model. Furthermore, we use this technique to obtain the equilibrium workload and response time distribution to study redundancy with identical replicas and compare it to redundancy with independent replicas. For independent replicas we rely on the method suggested in [6] to obtain the equilibrium workload and response time distribution which also provides a DIDE, we introduce it in our setting along with redundancy with identical replicas in order to illustrate the similarities/differences in the approach taken to derive it. This DIDE is also a DDE for discrete and an IDE for continuous job sizes. When job sizes are PH-distributed, we can again simplify the associated IDE to an ODE.

Our main findings in case of identical replicas can be summarized as follows: When identical replicas are used, the stability region shrinks severely as d increases and depends on the higher moments of the job size distribution. As such replicating too much can easily cause system instability. More variable job size distributions tend to result in a larger stability region, but still cause larger response times when the system load is low. The mean and the variance of the response time in a system with redundancy typically remains low and increases sharply as the system gets close to becoming unstable. This increase is considerably sharper than in a system without redundancy. The mean response time tends to increase linearly with the squared coefficient of variation (SCV) of the job size distribution. For small SCVs increasing the SCV may reduce the mean response time. Finally, the tails of the response time distribution often decay much faster compared to a system without redundancy. We further show that these insights are considerably different from what is observed in a system with independent replicas, where the stability region often increases as more replicas are used, the mean response time tends to decrease as the SCV increases, etc.

The models considered in the paper (redundancy with independent resp. identical replicas) are introduced in Section 2. The cavity processes associated to these queueing systems is presented in Section 3. The DIDEs are derived in Section 4 where we also take a closer look at the numerical method used to compute the equilibrium workload distribution. In Section 5 we show the accuracy of our suggested method by comparing with simulations of finite dimensional systems and validate our method to obtain the stability region by means of simulation. Numerical results on redundancy with identical replicas can be found in Section 6. In Section 7 we make a comparison between having independent and identical replicas. Section 8 discusses some future work.

2 MODEL DESCRIPTION

We consider a system with N identical servers (for large N), each having an infinite waiting room. Arrivals occur according to a Poisson process with rate λN. The service discipline at each server is assumed to be first-come-first-served (FCFS) and jobs are processed at a constant rate 1. The job sizes are distributed with cumulative distribution function (cdf) $G(\cdot)$, complementary cdf (ccdf) $\bar{G}(\cdot)$ and mean $\mathbb{E}[G]$. We assume the job size distribution has no atom in zero, i.e. $G(0) = 0$. We can



(a) In dark gray, the workload right before the potential arrival occurs, in light gray the amount of work that arrives due to the potential arrival.

(b) In dark gray, the workload right after the potential arrival occurs, i.e. all light gray area right of the dotted line in Figure 1a.

Fig. 1. Graphical representation of what happens at an arrival instant for Red$_{\text{eq}}$(d) with $d = 3$ and $N = 5$.

decompose G in a continuous and discrete part by stating $G = G_1 + G_2$, where G_1 has a density function $g_1 : [0, \infty) \longrightarrow [0, \infty)$, G_2 has a discrete density given by $g_2(\cdot) = \sum_{n=0}^{\infty} p_n \delta_{a_n}(\cdot)$ (assuming $0 < a_0 < a_1 < \ldots$) and we have $\int_0^\infty g_1(u)\,du + \sum_{n=0}^{\infty} p_n = 1$. Here we use the notation δ_x for the dirac measure. For simplicity we denote $g(s)\,ds = g_1(s)\,ds + g_2(s)$, thus for a subset $A \subseteq [0, \infty)$ we have $\int_A g(s)\,ds = \int_A g_1(s)\,ds + \sum_n p_n \delta_{a_n}(A)$, where $\delta_a(A)$ equals one if $a \in A$ and zero otherwise. By abuse of notation, for $s \in (0, \infty)$ we write $g(s) = g_1(s) + \sum_{n=0}^{\infty} p_n \delta\{s = a_n\}$ (where $\delta\{x = y\}$ equals one if $x = y$ and zero otherwise). In what follows we generally employ the following notation: a capital letter for cdf, a capital letter with an overline for ccdf, a lowercase letter for pdf and $\mathbb{E}$ for an expectation.

For this model we consider 2 distinct policies:

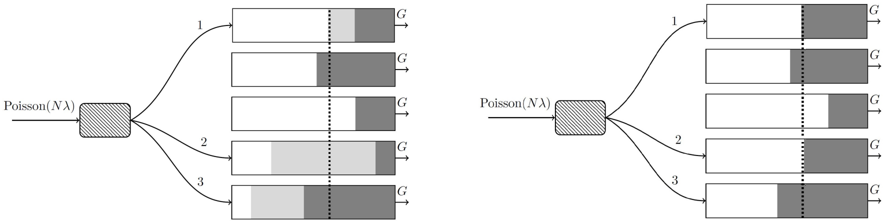
Redundancy-d with identical replicas (Red$_{\text{eq}}$(d)) : Each incoming job is replicated d times and each replica joins a random server (in total d, distinct, random servers receive an identical arrival). As soon as one replica finishes service, the remaining replicas are cancelled (whether in service or not). Cancellation is assumed to be immediate, although this assumption can be relaxed (see Section 4.2). It is important to emphasize that in this model, all d replicas of one job are assumed to be identical (i.e. equal in size).

Redundancy-d with independent replicas (Red$_{\text{iid}}$(d)) : At each arrival instant, replicas are made and distributed in the same manner as for Red$_{\text{eq}}$(d). The processing times of the d replicas of a job are however i.i.d. rather than identical, this model was studied in [6].

In Figure 1, we graphically show what happens at an arrival instant for $N = 5$ and $d = 3$ in the Red$_{\text{eq}}$(d) model. The dark gray area indicates actual workload while the light gray area indicates potential workload from the arrival. The same arbitrary amount of work is added to all (randomly) selected servers, this work is indicated in light gray in Figure 1a. All work that still needs to be done after the shortest queue finished serving the job may be discarded and the actual new workload of the queues is depicted in dark gray in Figure 1b.

In Figure 2, we show the events at an arrival instant for $N = 5$ and $d = 3$ for the Red$_{\text{iid}}$(d) model. An independent arbitrary amount of work is added to each selected queue. The workload at each chosen queue is then increased to match the workload at the server that finishes the new job first (which is the first queue in this case).

As in [7] the corresponding Markov processes only need to keep track of the workload at each of the N queues. We provide an analysis for Red$_{\text{eq}}$(d), the policy Red$_{\text{iid}}$(d) has been studied in [6]. We restate their result for general job sizes in our notation in Proposition 4.6. Red$_{\text{eq}}$(d) is stable if $\lambda\mathbb{E}[G] < 1/d$ and unstable for $\lambda\mathbb{E}[G] \geq 1$, its stability is unclear for $\lambda\mathbb{E}[G] \in (1/d, 1)$. It was shown



(a) In dark gray, the workload right before the potential arrival occurs, in light gray the amount of work that arrives due to the potential arrival.

(b) In dark gray, the workload right after the potential arrival occurs, i.e. all light gray area right of the dotted line in Figure 2a is now dark gray.

Fig. 2. Graphical representation of what happens at an arrival instant for Red$_{\text{iid}}$(d) with $d = 3$ and $N = 5$.

in [6] that Red$_{\text{iid}}$(d) with exponential job sizes is stable iff $\lambda\mathbb{E}[G] < 1$, one would expect that the stability region grows as a function of the job size variability (note that, for deterministic job sizes, these policies are equivalent).

3 CAVITY PROCESS

The cavity process methodology introduced in [3] is used to analyze both systems. The cavity process intends to capture the evolution of the workload of one queue for the limiting system when the number of servers N tends to infinity.

- For Red$_{\text{eq}}$(d) we find that when a potential arrival of size S occurs to servers with workloads $U_1, \dots, U_d$, each workload U_i becomes $\max\{U_i, \min_{j=1}^d\{U_j\} + S\}$ after the (possibly redundant) work of the arrival has been added.
- For Red$_{\text{iid}}$(d) we find that when a potential arrival of i.i.d. sizes $S_1, \dots, S_d$ occurs to servers with workloads $U_1, \dots, U_d$, then each workload U_i becomes $\max\{U_i, \min_{j=1}^d\{U_j + S_j\}\}$.

Definition 3.1 (Cavity Process). Let $\mathcal{H}(t)$, $t \geq 0$, be a set of probability measures on $\mathbb{R}$ called the *environment process*. The *cavity process* $X^{\mathcal{H}(\cdot)}(t)$, $t \geq 0$, takes values in $\mathbb{R}$ 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 $X^{\mathcal{H}(\cdot)}(t)$ becomes $\max\left\{X^{\mathcal{H}(\cdot)}(t), \min_{j=2}^d\{X^{\mathcal{H}(\cdot)}(t) + S, U_j + S\}\right\}$ resp. $\max\left\{X^{\mathcal{H}(\cdot)}(t), \min_{j=2}^d\{X^{\mathcal{H}(\cdot)}(t) + S_1, U_j + S_j\}\right\}$ for Red$_{\text{eq}}$(d) resp. Red$_{\text{iid}}$(d). Here $U_2, \dots, U_d$ are $d - 1$ independent random variables with law $\mathcal{H}(t)$ and $S, S_1, \dots, S_d$ are $d + 1$ independent random variables with law G. 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 has distribution $\mathcal{H}(t)$ at time t:

Definition 3.2 (Equilibrium Environment). When a cavity process $X^{\mathcal{H}(\cdot)}(\cdot)$ 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 $X^{\mathcal{H}(\cdot)}(t)$ has distribution $\mathcal{H}$ for all t.

The modularized program for analyzing load balancing systems presented in [3] when applied to our policies involves the following steps (assuming stability for N large):

a. Asymptotic Independence. Demonstrate $\Pi^N \to \Pi$ as $N \to \infty$, where Π^N is the stationary distribution for the studied policy with N queues and Π is a stationary and ergodic distribution on $[0,\infty)^\infty$. Show that the limit Π is unique, depending only on the service time distribution. Show that, for every k:

$$\Pi^{(k)} = \bigotimes_{i=1}^{k} \Pi^{(1)},$$

where $\Pi^{(k)}$ is Π restricted to its first k coordinates.

b. The queue at the cavity. Let $\mathcal{B}_s^N$ denote the arrival size distribution (which may be zero with a non-zero probability) in case of a potential arrival when the queue at the cavity has workload s. Show that the arrival process of a queue in the system of size N converges to a Poisson process with rate λd and a job size distribution $\mathcal{B}_s$ that depends on the workload s at arrival time. Denote $\mathcal{B} = \{\mathcal{B}_s, s \geq 0\}$.

c. Calculations. Given $\mathcal{B}$, the arrival size distributions, analyze the queue at the cavity in the large N limit using queueing techniques to express $\Pi^{(1)}$ as a function of $\mathcal{B}$:

$$\Pi^{(1)} = T(\mathcal{B}).$$

The arrival size distribution is determined by the workload distribution $\Pi^{(1)}$ (as explained above) we thus have:

$$\mathcal{B} = H(\Pi^{(1)}).$$

We then must solve these two fixed point equations to obtain the equilibrium environment $\Pi^{(1)} = \mathcal{H}$.

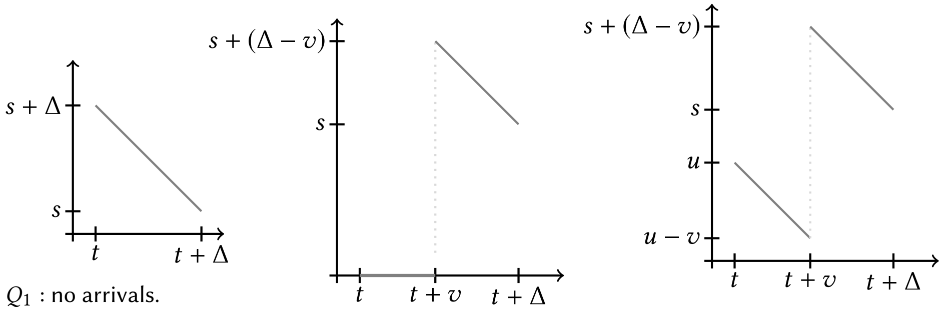
In this work, we focus on **c**, the computational step of the program. We present a numerical method to compute the Equilibrium Environment $\mathcal{H}$ corresponding to Red$_{\text{eq}}$(d), and validate it with simulation. Under the same setup, Red$_{\text{iid}}$(d) can be studied. Therefore we conjecture (numerical evidence to support this conjecture is presented in Section 5):

Conjecture 3.3. *Consider a load balancing system operating under the Red$_{\text{eq}}$(d) or Red$_{\text{iid}}$(d) policy on N servers, assume λ, d and G are such that this system is uniformly stable for sufficiently large N and the local service is FCFS. Then, in the large N limit, there is a unique equilibrium distribution. Under this distribution, any finite number of queues are independent. Moreover, this equilibrium can be found as the unique fixed point in step **c**.*

We now characterize the evolution of the cavity process associated with the equilibrium environment process. Let $f(t,s)$, $t \in [0,\infty)$, $s \in (0,\infty)$ describe the density at which a random server, at time t, has workload $s > 0$. Note that $f(t,\cdot)$ is not an actual pdf as the probability that the server is empty is non-zero. Let $F(t,s) = F(t,0) + \int_0^s f(t,u)du$ denote the cdf of the workload of a random server, here $F(t,0) = 1 - \int_0^\infty f(t,s)$ is the probability that a random server is idle.

We define $c_d(t,s,r)$ as the double density that, if a potential arrival occurs at time t, the queue at the cavity has workload $s > 0$ and its workload is increased to $r > s$ by the potential arrival. Lastly we let $C_d(t,r)$ denote the density at which, if a potential arrival occurs at time t, the queue at the cavity has workload 0 and its workload is increased to $r > 0$.

We now obtain a partial DIDE (PDIDE) which describes the transient evolution of the cavity queue as a function of c_d, C_d. The proof is similar to the proof of Theorem 3.4 in [7].



(a) Q_1 : no arrivals.

(b) Q_2 : arrival to empty queue.

(c) Q_3 : arrival to non-empty queue.

Fig. 3. Graphical representation to illustrate (3), all ways one can have workload s at time $t+\Delta$ (which are not $o(\Delta)$).

Theorem 3.4. *The evolution of the cavity process associated to the equilibrium environment process of the Red$_{\text{eq}}$(d),Red$_{\text{iid}}$(d) policy is captured by the following set of equations:*

$$\frac{\partial f(t,s)}{\partial t} - \frac{\partial f(t,s)}{\partial s} = \lambda d \cdot \left(- \int_s^\infty c_d(t,s,r)dr + C_d(t,s) + \int_0^s c_d(t,u,s)du \right) \tag{1}$$

$$\frac{\partial F(t,0)}{\partial t} = -\lambda d F(t,0) + f(t,0^+), \tag{2}$$

for $s > 0$, where $f(x,z^+) = \lim_{y \downarrow z} f(x,y)$.

Proof. We first let $t, s > 0$ and $0 < \Delta < s$ be arbitrary. We now describe the possible evolution of the workload of the queue at the cavity in the interval $[t, t+\Delta]$ s.t. it has exactly workload s at time $t+\Delta$. We write:

$$f(t+\Delta, s) = Q_1 + Q_2 + Q_3 + o(\Delta), \tag{3}$$

and describe how to obtain these Q_i.

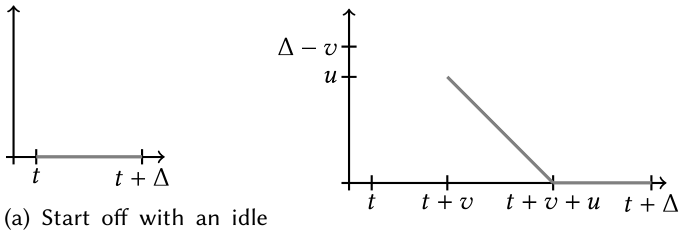
(Q_1) First, we consider the case where the queue at the cavity has $s+\Delta$ work at time t and no potential arrivals in $[t, t+\Delta]$ make its workload increase. For this case we find: $Q_1 = f(t, s+\Delta) - \lambda d \int_0^\Delta \int_{s+\Delta-v}^\infty c_d(t+v, s+\Delta-v, r)drdv$.

(Q_2) Second, we consider the case in which the queue at the cavity is empty at time $t+v$, $v \in [0,\Delta]$ and its workload is increased to $s+(\Delta-v)$ by a potential arrival. This happens with density: $Q_2 = \lambda d \int_0^\Delta C_d(t+v, s+(\Delta-v))dv$.

(Q_3) Lastly, the queue at the cavity may be non-empty at time $t+v$, $v \in [0,\Delta]$ and its workload increases to $s+(\Delta-v)$ by a potential arrival. This case has density: $Q_3 = \lambda d \int_0^\Delta \int_v^{s+\Delta} c_d(t+v, u-v, s+(\Delta-v))dudv$.

We graphically show the three options, Q_1, Q_2 and Q_3 in Figure 3. Note that any other event involves having at least 2 arrivals which yields terms that are $o(\Delta)$. Subtracting $f(t, s+\Delta)$, dividing by Δ and taking the limit $\Delta \to 0$ on both sides of (3), we find that (1) indeed holds.

We have not yet considered the case $s = 0$, for this we need to consider which events on $[t, t+\Delta]$ result in the workload of the queue at the cavity to be 0 at time $t+\Delta$. To this end we consider the following scenarios:



(a) Start off with an idle server.

(b) Start off with a non-idle server.

Fig. 4. Graphical representation to illustrate all ways one can end up with an empty queue at time $t + \Delta$ (which are not $o(\Delta)$).

- The queue at the cavity is empty at time t and no actual arrivals occur in the interval $[t, t + \Delta]$. As each potential arrival is an actual arrival for empty queues, we find that this event occurs with probability $F(t, 0)(1 - \lambda d\Delta)$.
- The queue at the cavity is non empty at some time $t + v, v \in [0, \Delta]$, and decreases to zero by time $t + \Delta$. We find that this event occurs with probability $\int_0^\Delta \int_0^{\Delta - v} f(t + v, u)\, du\, dv$.

Putting these together, we find that the following equality holds for $F(t + \Delta, 0)$:

$$F(t + \Delta, 0) = F(t, 0)(1 - \lambda d\Delta) + \int_0^\Delta \int_0^{\Delta - v} f(t + v, u) du dv + o(\Delta).$$

Subtracting $F(t, 0)$, dividing by Δ and taking the limit $\Delta \longrightarrow 0$ on both sides results in (2). □

Remark. *The PDIDE found in Theorem 3.4 could alternatively have been derived using the generalized Master Equation given by (7.25-7.26) in [10].*

We still require an exact expression for c_d and C_d. Moreover, we need an efficient method to compute the quantities $\int_s^\infty c_d(t, s, r)\, dr$ and $\int_0^s c_d(t, u, s)\, du$. Therefore, in the next Proposition, we describe how to determine c_d and C_d.

Proposition 3.5. *For the* Red$_{\text{eq}}(d)$ *policy we have* $c_d(t, s, r) = c_{d,1}(t, s, r) + c_{d,2}(t, s, r) + c_{d,3}(t, s, r)$ *such that:*

$$\int_s^\infty c_{d,1}(t, s, r) dr = \bar{G}(s) f(t, s)(1 - \bar{F}(t, 0)^{d-1}) \tag{4}$$

$$\int_s^\infty c_{d,2}(t, s, r) dr = f(t, s)\bar{F}(t, s)^{d-1} \tag{5}$$

$$\int_s^\infty c_{d,3}(t, s, r) dr = (d - 1) f(t, s) \left(\bar{F}(t, \cdot)^{d-2} f(t, \cdot) * \bar{G}(\cdot)\right)(s) \tag{6}$$

$$\int_0^s c_{d,1}(t, u, s) du = g(s) \cdot (F(t, s) - F(t, 0))(1 - \bar{F}(t, 0)^{d-1})$$

$$\int_0^s c_{d,2}(t, u, s) du = \left(g(\cdot) * f(t, \cdot)\bar{F}(t, \cdot)^{d-1}\right)(s)$$

$$\begin{aligned}\int_0^s c_{d,3}(t, u, s) du = {} & (d - 1)F(t, s) \cdot \left(g(\cdot) * f(t, \cdot) \cdot \bar{F}(t, \cdot)^{d-2}\right)(s) \\ & - (d - 1)\left(g(\cdot) * F(t, \cdot) f(t, \cdot)\bar{F}(t, \cdot)^{d-2}\right)(s),\end{aligned}$$

where $(f_1 * f_2)(s) = \int_0^s f_1(u)f_2(s-u)du$ denotes the convolution product. These quantities can all be computed quickly which simplifies solving (1-2) significantly. Lastly, we have $C_d(t,s) = F(t,0)\cdot g(s)$.

Proof. First we define $c_{d,1}, c_{d,2}$ and $c_{d,3}$ as follows:

- At least one of the $d-1$ independent random variables with law $\mathcal{H}(t)$ is zero and the incoming job has size r. We find (for $s < r$):
$$c_{d,1}(t,s,r)\,dr\,ds = g(r)f(t,s)(1-\bar{F}(t,0)^{d-1})\,dr\,ds.$$
- The queue at the cavity is the queue with the minimal workload (i.e. s) and the size of the arrival is exactly $r-s$:
$$c_{d,2}(t,s,r)\,dr\,ds = g(r-s)f(t,s)\bar{F}(t,s)^{d-1}\,dr\,ds.$$
- The queue with minimal workload has $0<u<s$ workload, where s is the workload of the queue at the cavity, and the arrival size is $r-u$:
$$c_{d,3}(t,s,r)\,dr\,ds = (d-1)f(t,s)\int_0^s g(r-u)\bar{F}(t,u)^{d-2}f(t,u)\,du\,dr\,ds.$$

now the claimed equalities all follow from direct computation and applying Fubini (which is allowed as all integrands are positive functions). It is trivial to derive the expression for C_d. □

The PDIDE (1-2) can now be solved using an (improved) Euler scheme. This result is also of interest to obtain a fixed point equation for the equilibrium environment, i.e., workload distribution. In the subsequent section, we provide an efficient method to compute the equilibrium workload (and thus also response time) distribution.

4 EQUILIBRIUM REGIME

For the equilibrium we use the same notations as in the transient case, but we leave out the time dependence, i.e., we write $f(s)$ instead of $f(t,s)$ and set $\frac{\partial f(t,s)}{\partial t} = 0$. From the PDIDE describing the transient behaviour (1-2) we now derive a method to compute the equilibrium workload distribution.

Proposition 4.1. *The equilibrium workload distribution associated to the equilibrium environment of $Red_{\text{eq}}(d)$ or $Red_{\text{iid}}(d)$ satisfies the following equation:*

$$\bar{F}'(s) = -f(s) = -\lambda d\left(F(0)\bar{G}(s) + \int_0^s\int_s^\infty c_d(u,v)dvdu\right) \tag{7}$$

Proof. Integrating (1) w.r.t. s and using (2) as a boundary condition ($f(0^+) = \lambda dF(0)$), we obtain:

$$f(s) = \lambda d\Bigg(F(0) - \int_0^s C_d(u)du + \int_0^s\int_u^\infty c_d(u,r)\,dr\,du - \int_0^s\int_0^r c_d(u,r)\,du\,dr\Bigg). \tag{8}$$

We can further simplify (8) by applying Fubini to the term $\int_0^s\int_0^r c_d(u,r)\,du\,dr$ to obtain (7). □

4.1 Red_{eq}(d)

From Proposition 4.1 we obtain a simple DIDE which can be solved numerically:

Theorem 4.2. *The stationary workload distribution associated to the equilibrium environment satisfies the following DIDE:*

$$\bar{F}'(s) = -\lambda d\left[\bar{G}(s)(1-\bar{F}(s)) + \int_0^s g(u)\bar{F}^{d-1}(s-u)(\bar{F}(s-u)-\bar{F}(s))\,du\right]. \tag{9}$$

Proof. Using (4-6) we can simplify (7) applied to Red_{eq}(d) to obtain:

$$\begin{aligned}\bar{F}'(s) = -\lambda d\Big[&\bar{G}(s)\left((1-\bar{F}(0)^d) - \bar{F}(s)(1-\bar{F}(0)^{d-1})\right) \\ &+ d(\bar{G} * f\bar{F}^{d-1})(s) - (d-1)\bar{F}(s)(\bar{G} * f\bar{F}^{d-2})(s)\Big] \qquad (10)\end{aligned}$$

We decompose the ccdf $\bar{G} = \bar{G}_1 + \bar{G}_2$, where $\bar{G}_1$ corresponds to the continuous part and $\bar{G}_2$ the discrete part. For the continuous part we apply integration by parts to obtain:

$$(\bar{G}_1 * f\bar{F}^n)(s) = \frac{1}{n+1}\left(\bar{G}_1(s)\bar{F}^{n+1}(0) - \bar{G}_1(0)\bar{F}^{n+1}(s) + (g_1 * \bar{F}^{n+1})(s)\right). \qquad (11)$$

For the discrete part we first define $\iota(s) = \sup\{n \mid a_n \le s\}$, where a_n for $n \in \{1, 2, \dots\}$ are the atoms of the job size distribution. We find that the following equality holds:

$$(\bar{G}_2 * f\bar{F}^n)(s) = \frac{1}{n+1}\left(\bar{G}_2(s)\bar{F}^{n+1}(0) - \bar{G}_2(0)\bar{F}^{n+1}(s) + \sum_{n=0}^{\iota(s)} p_n\bar{F}^{n+1}(s-a_n)\right). \qquad (12)$$

This equation follows by splitting the integral over the intervals $[0, a_0]$, $[a_0, a_1]$, $\dots$, $[a_{\iota(s)-1}, a_{\iota(s)}]$, $[a_{\iota(s)}, s]$. Putting integrands together and using the definition of g, we find that (11-12) simplifies to (9). □

The ccdf of the waiting time is given by $\bar{F}_W(s) = \bar{F}(s)^d$. The ccdf of the response time is given by the convolution of g with the ccdf of the waiting time, i.e. $\bar{F}_R(s) = \bar{G}(s) + (g * \bar{F}_W)(s)$.

The DIDE found in (9) can be simplified to a set of ODEs in case job sizes have a PH distribution.

Corollary 4.3. *If job sizes have a PH distribution with parameters (α, A) then (9) simplifies to the following ODE:*

$$\begin{aligned}\bar{F}'(s) &= -\lambda d\alpha\left[e^{sA}\mathbf{1}(1-\bar{F}(s)) + h_1(s) - h_2(s)\bar{F}(s)\right] \\ h_1'(s) &= Ah_1(s) + \mu\bar{F}^d(s) \\ h_2'(s) &= Ah_2(s) + \mu\bar{F}^{d-1}(s),\end{aligned}$$

with $\mu = -A\mathbf{1}$ and boundary condition $h_1(0) = h_2(0) = 0$.

Proof. For PH distributed job sizes we find $\bar{G}(s) = \alpha e^{sA}\mathbf{1}$ and $g(s) = \alpha e^{sA}\mu$. Applying this to (9) and splitting terms, we find:

$$\bar{F}'(s) = -\lambda d\alpha\left[e^{sA}\mathbf{1}(1-\bar{F}(s)) + \int_0^s e^{uA}\mu\bar{F}^d(s-u)\,du - \int_0^s e^{uA}\mu\bar{F}^{d-1}(s-u)\,du\bar{F}(s)\right]. \qquad (13)$$

Letting $h_1(s) = \int_0^s e^{uA}\mu\bar{F}^d(s-u)\,du = \int_0^s e^{(s-u)A}\mu\bar{F}^d(u)\,du$ we find:

$$h_1'(s) = Ah_1(s) + \mu\bar{F}^d(s).$$

Analogously for $h_2(s) = \int_0^s e^{uA}\mu\bar{F}^{d-1}(s-u)\,du$ we find $h_2'(s) = Ah_2(s) + \mu\bar{F}^{d-1}(s)$. □

Remark. *We can generalize the result in Corollary 4.3 to conditioned Phase type distributions. I.e. let X have PH distribution with parameters (α, A) and $a < b$ two positive numbers. If job sizes have distribution $(X \mid a < X < b)$ we find:*

$$\bar{G}(s) = \begin{cases} 1 & s < a \\ p\alpha(e^{As} - e^{Ab})\mathbf{1} & a \le s < b \\ 0 & b \le s \end{cases}, g(s) = \begin{cases} 0 & s < a \\ p\alpha e^{As}\mu & a \le s < b \\ 0 & b \le s \end{cases}$$

with $p = \frac{1}{\alpha(e^{Aa}-e^{Ab})\mathbf{1}}$. This allows us to find the following delayed differential equation for the equilibrium workload distribution:

$$\begin{aligned}
\bar{F}'(s) &= -\lambda d(1-\bar{F}(s)) && s \le a\\
\bar{F}'(s) &= -\lambda d\left[p\alpha(e^{As}-e^{Ab})\mathbf{1}(1-\bar{F}(s)) + p\alpha(h_1(s)-h_2(s)\bar{F}(s))\right] && a < s \le b\\
\bar{F}'(s) &= -\lambda d p\alpha(h_1(s)-h_2(s)\bar{F}(s)) && b < s,
\end{aligned}$$

where h_1, h_2 satisfy $h_1(a) = h_2(a) = 0$ and

$$\begin{aligned}
h_1'(s) &= Ah_1(s) + e^{Aa}\mu\bar{F}^d(s-a) && a < s \le b\\
h_1'(s) &= Ah_1(s) + (e^{Ab}\bar{F}^d(s-b) - e^{Aa}\bar{F}^d(s-a))\mu && b < s\\
h_2'(s) &= Ah_2(s) + e^{Aa}\mu\bar{F}^{d-1}(s-a) && a < s \le b\\
h_2'(s) &= Ah_2(s) + (e^{Ab}\bar{F}^{d-1}(s-b) - e^{Aa}\bar{F}^{d-1}(s-a))\mu && b < s.
\end{aligned}$$

Remark. *It is not hard to see that when the job size distribution is some combination (product, sum, mixture, . . .) of discrete and PH-distributed random variables, one can still obtain a DDE by generalizing Corollary 4.3.*

4.2 Red$_{\mathbf{eq}}$(d) with delayed cancellation

We now assume there is some delay in the cancellation of jobs, i.e. after the first server finishes a job, the other $d-1$ servers continue working on the job for some time $\delta > 0$. We find the following result:

Proposition 4.4. *The stationary workload distribution associated to the equilibrium environment for Red$_{\text{eq}}$(d) with a cancellation delay equal to $\delta > 0$ satisfies the following DIDE:*

$$\bar{F}'(s) = -\lambda d\left(\bar{G}(s) + \int_0^s g(s-u)\bar{F}(u)\,du - \bar{F}(s)\right) \qquad s \le \delta \tag{14}$$

$$\begin{aligned}
\bar{F}'(s) = -\lambda d\Bigg(&\bar{G}(s) - \bar{F}(s)\bar{G}(s-\delta) + \int_0^\delta \bar{F}(u)g(s-u)\,du\\
&+ \int_0^{s-\delta} \bar{F}(s-u-\delta)^{d-1}(\bar{F}(s-u)-\bar{F}(u))g(u)\,du\Bigg) \qquad s > \delta.
\end{aligned} \tag{15}$$

Proof. It is not hard to see (analogue to Proposition 3.5) that in this case we have:

$$\begin{aligned}
c_d(s,r)\,dr\,ds &= g(r-\delta)f(s)(1-\bar{F}(0)^{d-1})\,dr\,ds\\
&\quad + g(r-s)f(s)\bar{F}(s)^{d-1}\,dr\,ds\\
&\quad + (d-1)f(s)\int_0^s g(r-u-\delta)\bar{F}(u)^{d-2}f(u)\,du\,dr\,ds.
\end{aligned}$$

The result then follows using arguments similar to the proof of Theorem 4.2. □

Remark. *As for the Red$_{eq}$(d) model, the ccdf of the waiting time is given by $\bar{F}_W(s) = \bar{F}(s)^d$ and the response time by $\bar{F}_R(s) = \bar{G}(s) + (g * \bar{F}_W)(s)$.*

The above DIDE simplifies to a delayed differential equation in case X has a PH distribution:

Corollary 4.5. *If job sizes have a PH distribution with parameters (α, A), then the DIDE given by (14-15) simplifies to the following DDE:*

$$\begin{aligned}
\bar{F}'(s) &= -\lambda d(\bar{G}(s) + \alpha\xi_1(s) - \bar{F}(s)) && s \leq \delta \\
\bar{F}'(s) &= -\lambda d\Big(\bar{G}(s) - \bar{F}(w)\bar{G}(w-\delta) + \alpha(\xi_1(s) + \xi_2(s) - \xi_3(s)\bar{F}(s))\Big) && s > \delta \\
\xi_1'(s) &= A\xi_1(s) + \bar{F}(s)\mu && s \leq \delta \\
\xi_1'(s) &= A\xi_1(s) && s > \delta \\
\xi_2'(s) &= \bar{F}(w-\delta)^{d-1}\bar{F}(s)\mu + A\xi_2(s) && s > \delta \\
\xi_3'(s) &= \bar{F}(w-\delta)^{d-1}\mu + A\xi_3(s) && s > \delta,
\end{aligned}$$

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

Proof. This follows from Proposition 4.4 by defining :

$$\begin{aligned}
\xi_1(s) &= \int_0^{\min\{s,\delta\}} e^{(s-u)A}\bar{F}(u)\,du\mu \\
\xi_2(s) &= \int_0^{s-\delta} F(u)^{d-1}\bar{F}(u+\delta)e^{(w-u-\delta)A}\,du\mu \\
\xi_3(s) &= \int_0^{s-\delta} \bar{F}(u)^{d-1}e^{(w-u-\delta)A}\,du\mu.
\end{aligned}$$

□

4.3 Red$_{\text{iid}}$(d)

If we were to analyze Red$_{\text{iid}}$(d) in the same manner as we did for Red$_{\text{eq}}$(d), we would again find that the ccdf of the workload distribution satisfies equation (7). In the case of Red$_{\text{iid}}$(d), we find for arbitrary $0 < s < r$:

$$c_d(s,r) = f(s) \cdot \left(g(r-s)\bar{F}_{U+S}(r)^{d-1} + (d-1)\bar{G}(r-s)f_{U+S}(r)\bar{F}_{U+S}(r)^{d-2}\right), \tag{16}$$

where U and S are random variables with distribution F and G. Plugging (16) in (7) one finds a functional differential equation describing the workload distribution. This equation is hard to solve and it is not immediately clear how to simplify it. However, as shown in [6] the following result holds (the proof of which we summarize in a few words, for more details see [6]):

Proposition 4.6. *The equilibrium workload distribution associated to Red$_{\text{iid}}$(d) satisfies the following DIDE:*

$$\bar{F}'(s) = -\lambda d\bar{F}_{R_1}(s)^{d-1}(\bar{F}_{R_1}(s) - \bar{F}(s)), \tag{17}$$

with $\bar{F}_{R_1}(s) = \bar{G}(s) + (g * \bar{F})(s)$, the probability that the response time is at least s if a job is sent to only 1 server.

Proof. If one were to send only one replica, this replica has a response time which is at least s if and only if either the job size is at least s or the job size is exactly $u < s$ and the workload of the queue to which the replica is sent is at least $s - u$. This shows that $\bar{F}_{R_1}(s) = \bar{G}(s) + (g * \bar{F})(s)$.

The probability that a potential arrival increases the workload of the queue at the cavity from a value under s to a value above s is equal to the probability that all $d - 1$ replicas which are sent to other queues have a response time which is at least s, the response time of the replica sent to the queue at the cavity is at least s and the queue at the cavity has a workload which is at most s. Note that the individual response times R_1 can be written as a product because of the independence assumption of both the workload processes and the job sizes. □

For PH-distributed job sizes we find a result which is similar to Corollary 4.3:

Corollary 4.7. *If job sizes have a PH distribution with parameters (α, A) then (17) simplifies to the following ODE:*

$$\bar{F}'(s) = -\lambda d(\bar{G}(s) + \alpha h(s))^{d-1} \left(\bar{G}(s) + \alpha h(s) - \bar{F}(s)\right),$$
$$h'(s) = Ah(s) + \bar{F}(s)\mu,$$

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

Proof. This follows in a similar manner as Corollary 4.3, but here we set $h(s) = \int_0^s e^{(s-u)A}\mu\bar{F}(u)\,du$. □

Remark. *One can again generalize this result to conditional PH-distributions and combinations of PH-distributions and discrete distributions.*

In case of independent replicas there are several possible definitions for the waiting time, that is, it could be the time until the first replica enters service or the time until the replica that first completes service enters service. As such we only consider the response time, the ccdf of which is:

$$\bar{F}_R(s) = \left(\bar{G}(s) + (g * \bar{F})(s)\right)^d \left(= \bar{F}_{R_1}(s)^d\right).$$

4.4 Numerical Considerations

Throughout all sections which contain numerical examples, we exclusively make use of job size distributions which have a mean equal to one. In particular we focus our attention to the following job size distributions:

- *Exponential Job Sizes :* Job sizes have an exponential distribution with mean equal to one.
- *Deterministic Job Sizes :* Job sizes are always equal to one.
- *Bounded Pareto Job Sizes :* Job sizes are bounded Pareto with lower bound 0.2, upper bound 72 and $\alpha = 1.1$, meaning $\mathbb{E}[G] = 1$ and $\mathbb{E}[G^2] = 10$.
- *Hyperexponential Job Sizes :* Job sizes are hyperexponential with two phases and balanced means, chosen such that $\mathbb{E}[G] = 1$. When the Squared Coefficient of Variation SCV is not specified, we take $\mathbb{E}[G^2] = 10$.
- *Erlang Job Sizes :* Job sizes which have an Erlang distribution with 2 up to 50 phases such that $\mathbb{E}[G] = 1$ (i.e. $\lambda = k$ with k the number of phases).

Remark. *For bounded Pareto job sizes, we need to resort to solving an IDE which is $O(M^2)$, for all other job size distributions the required computation time is only $O(M)$. Here M denotes the number of control points used to numerically represent $\bar{F}$.*

Note that Theorem 4.2, Corollary 4.3, Proposition 4.6 and Corollary 4.7 do not specify a boundary condition for $\bar{F}(0)$. This is not surprising as $\bar{F}(0)$ corresponds to the unknown actual system load. We have the following Lemma which is used as a basis for an algorithm to find $\bar{F}(0)$:

Lemma 4.8. *Let $\lambda > 0, d \in \{2, 3, \dots\}$ be such that the associated system is stable then the following are equivalent:*

- *$\bar{F}(s)$ is a solution to (9) resp. (17) and $\inf_{s>0} \bar{F}(s) = 0$,*
- *$\bar{F}(s)$ is the unique ccdf of the workload equilibrium for $Red_{\mathrm{eq}}(d)$ resp. $Red_{\mathrm{iid}}(d)$.*

Proof. Obviously, if $\bar{F}(s)$ is the ccdf of the workload equilibrium, it is a solution to the associated fixed point equation and thus also of the associated DIDE, moreover $\inf_{s>0}\bar{F}(s) = 0$ holds for any ccdf.

We should still show that the solution $\bar{F}$ is indeed a ccdf if it is a solution to (9) or (17) and $\inf_{s>0}\bar{F}(s) = 0$. The uniqueness then follows from the Conjecture 3.3. We show that for arbitrary $t > 0$, we have: if for all $s < t, \bar{F}'(s) \leq 0$ and $\bar{F}(s) \geq 0$ then $\bar{F}'(t) \leq 0$, this shows that $\bar{F}$ is a decreasing function as it is positive. For (9) it suffices to note that $\bar{F}(t-u) - \bar{F}(t)$ appearing in the integral equals $-\int_{t-u}^{t} \bar{F}'(v)\,dv \geq 0$. For (17) we note that:

$$\begin{aligned}
F_{R_1}(t) &= \bar{G}(t) + \int_0^t \bar{F}(u)g(t-u)\,du \\
&\geq \bar{G}(t) + \int_0^t \bar{F}(t)g(t-u)\,du \\
&= \bar{G}(t) + G(t)\bar{F}(t) \\
&\geq \bar{F}(t).
\end{aligned}$$

We have thus shown that if $\bar{F}$ satisfies the stated conditions, it is also a non-increasing function, as $\inf_{s>0}\bar{F}(s) = 0$ we find that $\lim_{s\to\infty}\bar{F}(s) = 0$. This shows that $\bar{F}$ is indeed a ccdf. □

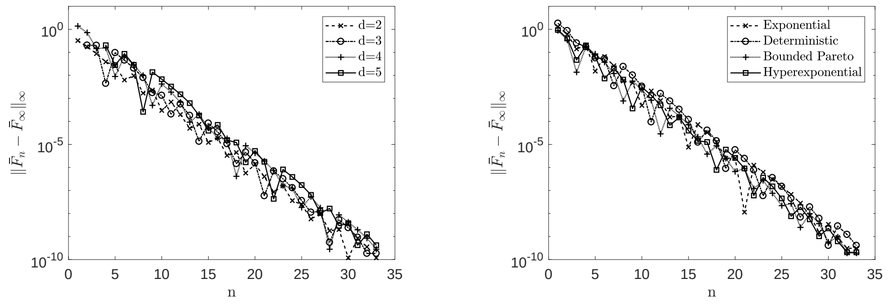
Based on the result shown in Lemma 4.8 we now obtain an algorithm which can be used to find the ccdf $\bar{F}$ which is the solution for (9) and (17). Also, we present a simple method to check whether for a given job size distribution, λ and d the system is stable (i.e. the equilibrium workload distribution is not infinite). Note that the DIDE however still makes sense when the system is unstable: we find the boundary condition $\bar{F}(0) = 1$ and from this it is not hard to see that both for Red$_{\text{eq}}$(d) and Red$_{\text{iid}}$(d) we find $\bar{F}(s) = 1$ for all s, i.e. the system load is almost surely infinite.

We employ the following bisection algorithm to find the value of $\bar{F}(0)$ for which the associated solution satisfies $\inf_{s>0}\bar{F}(s) = 0$:

(1) Set lb = 0 and ub = 1,
(2) Compute $y = \inf_{s>0}\bar{F}(s)$, where $\bar{F}(s)$ is computed as the solution of (9) or (17), with boundary condition $\bar{F}(0) = x_0 = \frac{\text{lb}+\text{ub}}{2}$.
(3) Set lb $= x_0$ if $y < 0$ otherwise set ub $= x_0$, return to Step 2.

Due to Lemma 4.8, we are certain this algorithm converges, provided that $\inf_{s>0}\bar{F}(s)$ is increasing as a function of the boundary condition $\bar{F}(0)$. Actually all we need is if $\bar{F}_1(0) < \bar{F}_2(0)$ and $\inf_{s>0}\bar{F}_2(s) < 0$ then $\inf_{s>0}\bar{F}_1(s) < 0$. Unfortunately this statement appears to be hard to prove and we only managed to confirm this numerically (see also Figure 7).

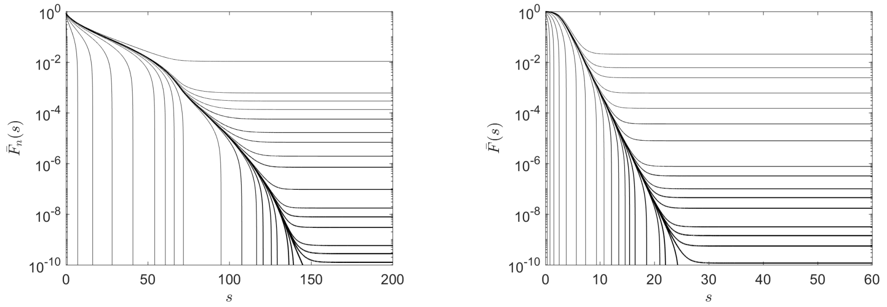
We now provide some deeper insight into how well the algorithm performs. For this discussion let us focus on Red$_{\text{eq}}$(d) and note that the discussion for Red$_{\text{iid}}$(d) is completely analogous. We need to numerically solve (9) for each step of the algorithm, which takes $O(N^2)$ resp. $O(N)$ time for continuous resp. discrete or PH distributed job sizes (use Corollary 4.3). For a function f, we let $\|f\|_\infty = \sup_{s>0}|f(s)|$ denote its supremum norm. In Figure 5, we first compute the limiting distribution $\bar{F}_\infty$ which satisfies (9) and $\bar{F}(0)$ is chosen such that $|\inf_{s>0}\bar{F}(s)| < 10^{-10}$. We let $\bar{F}_n$ denote the solution to (9) after n steps have been taken in the algorithm. We show the difference $\|\bar{F}_n - \bar{F}_\infty\|_\infty$ for n varying from 1 to 34 (note that 34 is the first value n for which $2^{-n} < 10^{-10}$). Figure 5a is for exponential job sizes, $\lambda = 0.48$ and $d = 2, 3, 4, 5$ while Figure 5b is for $d = 2$, $\lambda = 0.7$ and varying (i.e. exponential, deterministic, bounded Pareto and hyperexponential) job sizes. We observe that the accuracy of $\bar{F}_n$ increases exponentially which means that $\|\bar{F}_n - \bar{F}_\infty\|_\infty \approx |\bar{F}_n(0) - \bar{F}_\infty(0)|$. We now show $\bar{F}_n$ for $n = 1, \ldots, 34$ in Figure 6 (on a logarithmic scale, negative values are discarded). Rather than labelling each line, we increase the linewidth as n increases. We clearly observe that as



(a) Exponential Job Sizes with $\lambda = 0.48$.

(b) Varying job sizes, $d = 2$ and $\lambda = 0.7$.

Fig. 5. Convergence of $\bar{F}_n$, the ccdf found after n steps to the limiting distribution ccdf $\bar{F}_\infty$.



(a) Bounded Pareto Job Sizes.

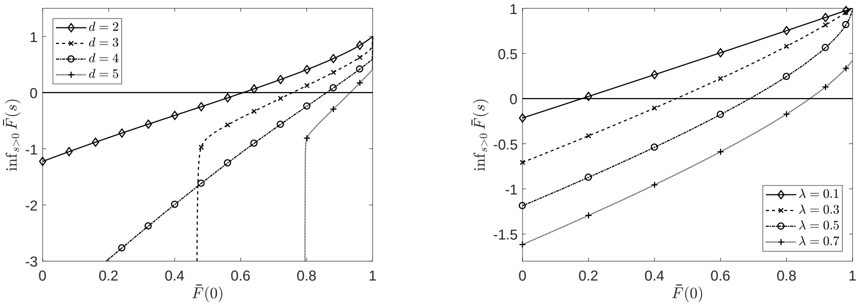
(b) Exponential Job Sizes.

Fig. 6. Plot of $\bar{F}_n$ with $n = 1, \dots, 34$, $\lambda = 0.7$ and $d = 2$ for bounded Pareto resp. exponential job size distributions. The linewidth is increased with n.

n increases, $\bar{F}_n$ gets increasingly closer to being an actual ccdf. Moreover we see that none of the lines cross, which supports the claim that if $\bar{F}_1(0) < \bar{F}_2(0)$, then for all $s : \bar{F}_1(s) < \bar{F}_2(s)$.

In Figure 7 we show $\inf_{s>0} \bar{F}(s)$, where $\bar{F}$ is the solution of (9) as a function of the used initial value $\bar{F}(0)$. In Figure 7a, job sizes are deterministic, $\lambda = 0.4$ and $d = 2, 3, 4, 5$. Here we observe that for certain values of λ, d and $\bar{F}(0)$ the infimum might be $-\infty$, but it is clearly monotone increasing and close to linear with a steeper ascend close to $\bar{F}(0) = 1$. In fact, all of these curves converge to 1 as $\bar{F}(0)$ converges to 1 and the incline close to 1 is so steep that it is not even visible in Figure 7. In Figure 7b, bounded Pareto job sizes are used with $d = 2$ and $\lambda = 0.1, 0.3, 0.5, 0.7$, we observe the same behaviour as for deterministic job sizes: the trajectory is close to linear with a steep incline when $\bar{F}(0)$ gets close to one. This curve, let us denote it by $y = f(x)$, is in fact the one for which we need to find the value x_0 that satisfies $f(x_0) = 0$ (i.e. this value x_0 is exactly the $\bar{F}(0)$ boundary condition for which $\bar{F}$ becomes a ccdf). The system is deemed unstable if and only if $y = f(x)$ does not cross zero in $[0, 1]$ and this happens when $f(x) < 0$ for all $x \in [0, 1)$, meaning there is a discontinuity in 1 as we necessarily have $f(1) = 1$ as mentioned before. Moreover, as $y = f(x)$ is a curve which is close to linear, better (i.e. faster) algorithms could be used to obtain a root of f, e.g. a simple Newton iteration would converge extremely fast in this case.

Let $\lambda_{\max}$ be defined as the highest value value of λ for which a system is still stable. That is, for any $\lambda < \lambda_{\max}$ the system is stable whilst for any $\lambda \geq \lambda_{\max}$ it is unstable. We know that for $\text{Red}_{\text{eq}}(\text{d})$,



(a) Deterministic Job Sizes and $\lambda = 0.4$.

(b) Bounded Pareto Job Sizes and $d = 2$.

Fig. 7. Plot of $\inf_{s>0} \bar{F}(s)$ as a function of $\bar{F}(0)$ for the solution of (9).

$\lambda_{\max} \in [1/d, 1]$. To obtain an algorithm which allows to compute $\lambda_{\max}$ we note that if we pick $\varepsilon > 0$ small and we find $\lambda > 0$ such that for this λ and $\bar{F}(0) = 1 - \varepsilon$ we have $\inf_{s>0} \bar{F}(s) = 0$, then the system load for this λ is $1 - \varepsilon$. If we now pick $\varepsilon > 0$ small enough this means that the system is close to instability, and this value can then be used as an approximation for $\lambda_{\max}$. We verify this method in Section 5.2 where we show that the approximation of $\lambda_{\max}$ obtained in this manner is at least accurate up to 0.001.

5 VALIDATION OF THE MODEL

5.1 Finite System Accuracy

In this section we use the algorithm presented in Section 4.4 to find the limiting workload distribution for $\text{Red}_{\text{eq}}(\text{d})$, where we find the value of $\bar{F}(0)$ such that $\bar{F}$ is a ccdf.

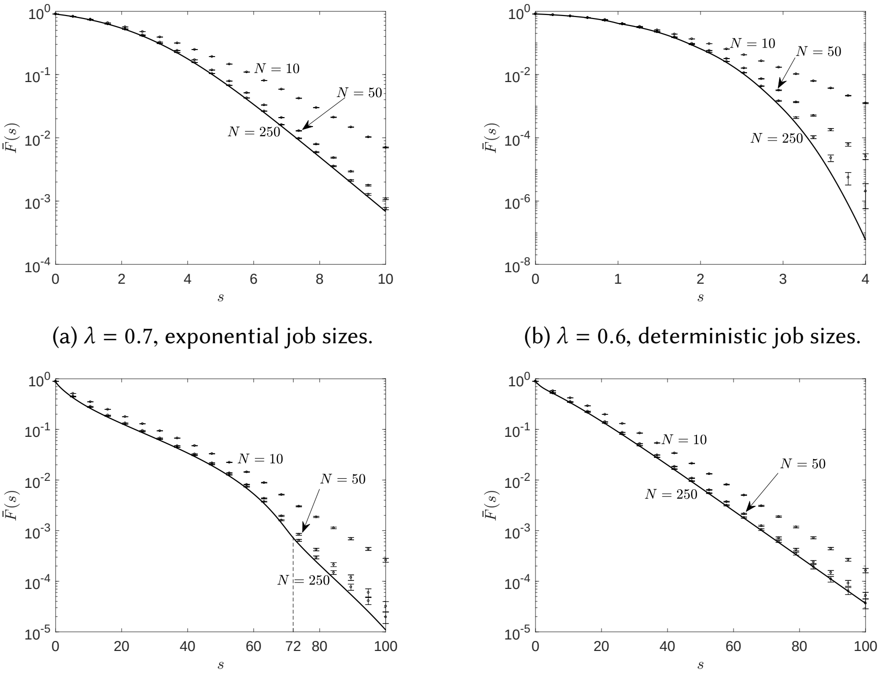
We compare the equilibrium workload distribution with the simulated workload distribution for a finite system with N servers. We do this for 4 of the main job size distributions: exponential, deterministic, bounded Pareto and hyperexponential. All simulation runs simulate the system up to time $t = 10^7/N$ and use a warm-up period of 30%. We simulate a system of $N = 10, 50, 250$ servers. The results are shown in Figure 8. We see that as N increases the approximation provided by the DIDE becomes more accurate (which supports Conjecture 3.3). Note that a similar figure can easily be made for the response time distribution and $\text{Red}_{\text{iid}}(\text{d})$.

5.2 Stability Region

In this section we apply the algorithm presented in Section 4.4 to obtain the maximum value $\lambda_{\max}$ for which the system is still stable. For this purpose, we first compute the value of $\lambda_{\max}$ for a system with $d = 2$ and deterministic, exponential resp. hyperexponential job sizes. We find:

- $\lambda_{\max} = 0.80554\ldots$ for deterministic job sizes,
- $\lambda_{\max} = 0.81669\ldots$ for exponential job sizes,
- $\lambda_{\max} = 0.83441\ldots$ for hyperexponential job sizes.

Note that in this example $\lambda_{\max}$ increases as the job size variability increases. For each distribution, we set $\lambda = \lambda_{\max} - 0.001$ and simulate a system with $N = 300$, arrival rate λ and the corresponding job size distribution for a time of $2 \cdot 10^4$. We observe in Figure 9a that the system indeed appears to be stable. In Figure 9b we observe that, when taking arrival rate $\lambda = \lambda_{\max} + 0.001$ and simulating the system with $N = 300$ seems to result in an unstable system for each job size distribution. This suggests that our method to obtain $\lambda_{\max}$ is indeed quite accurate.



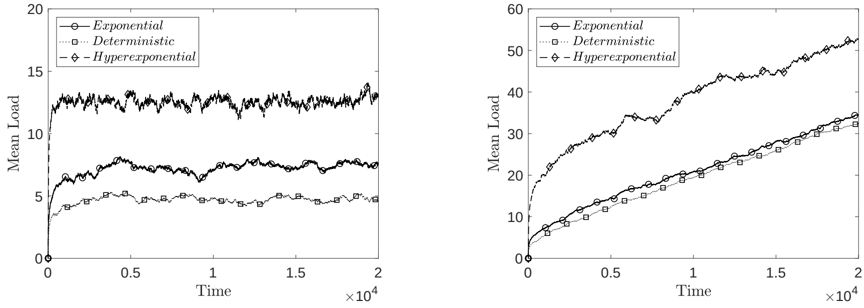
(a) $\lambda = 0.7$, exponential job sizes.

(b) $\lambda = 0.6$, deterministic job sizes.

(c) $\lambda = 0.7$, bounded Pareto job sizes (max=72).

(d) $\lambda = 0.7$, hyperexponential job sizes.

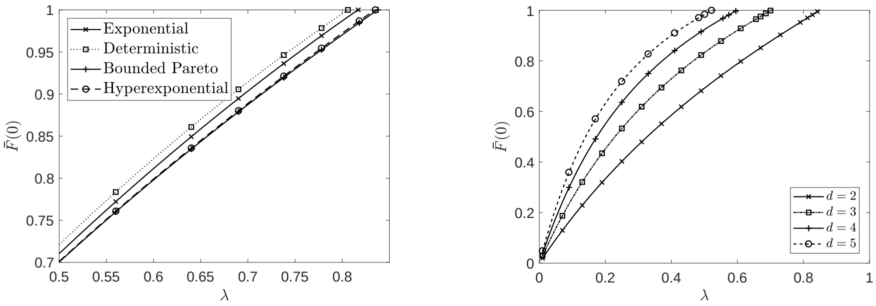
Fig. 8. For the Red$_{\text{eq}}$(d) policy: Limiting workload distribution vs. simulation for N servers with exponential, deterministic, bounded Pareto and hyperexponential job sizes. The full line represents the solution of the IDE/DDE/ODE, which is compared with the simulated 95% confidence intervals.



(a) Arrival rate $\lambda = \lambda_{\text{max}} - 0.001$, where λ_{max} depends on the job size distribution.

(b) Arrival rate $\lambda = \lambda_{\text{max}} + 0.001$, where λ_{max} depends on the job size distribution.

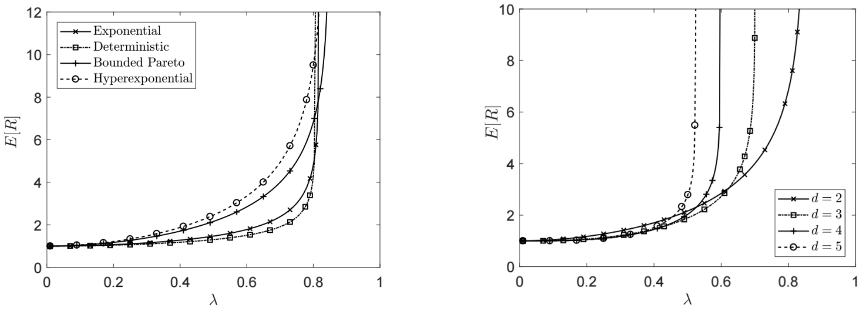
Fig. 9. The mean workload of a simulated system with $N = 300$ servers, $d = 2$ and deterministic, exponential resp. hyperexponential job sizes.



(a) $d = 2$ and different job size distributions.

(b) $d = 2, 3, 4, 5$ and bounded Pareto job sizes.

Fig. 10. Workload $\bar{F}(0)$ as a function of the arrival rate λ for Red$_{\text{eq}}$(d).



(a) $d = 2$ and different job size distributions.

(b) $d = 2, 3, 4, 5$ and bounded Pareto job sizes.

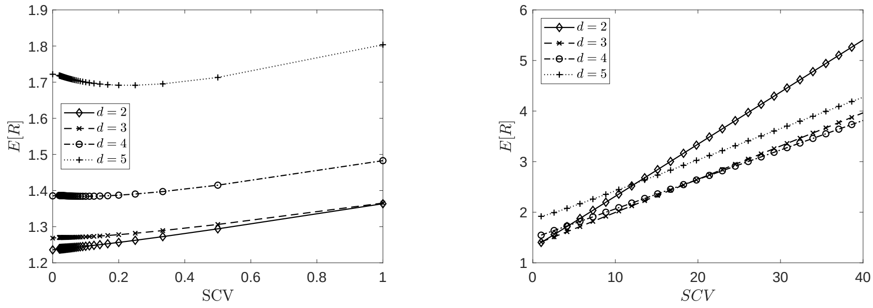
Fig. 11. Mean response time $\left(1 + \int_0^\infty \bar{F}(s)^d ds\right)$ as a function of the arrival rate λ for Red$_{\text{eq}}$(d).

6 NUMERICAL EXPERIMENTS FOR RED$_{\text{eq}}$(D)

6.1 Mean Response Time and Workload distribution

In Figures 10 and 11 we show the actual workload $\bar{F}(0)$ and the mean response time $\mathbb{E}[R] = 1 + \int_0^\infty \bar{F}(s)^d\, ds$ of the Red$_{\text{eq}}$(d) policy as a function of the arrival rate λ (recall $\mathbb{E}[G] = 1$). From Figure 10a, it is clear that the stability region not only depends on the mean and the variance of the job size distribution, but also on higher moments (as $\mathbb{E}[G^2] = 10$ for both the bounded Pareto and hyperexponential job sizes, see also Figure 14b). This makes the question of stability for Red$_{\text{eq}}$(d) for general job size distributions a hard problem (which in turn makes proving Conjecture 3.3 hard). We can already infer from the plot that the more variable the job size distribution, the lower the associated workload. From Figure 10b, it is obvious that $\lambda_{\max}$ (defined as the supremum of the arrival rates λ for which $\bar{F}(0) < 1$) decreases and the workload increases as a function of d (we have numerically verified that this also holds for the other job size distributions considered). For a more detailed discussion on $\lambda_{\max}$, see Section 6.3. Note that, as one would expect from a system that employs redundancy, the workload increases as a concave function as the arrival rate λ increases.

We show in Figure 11a that, despite the fact that the workload for the less variable jobs is consistently higher than that of the more variable ones, the same does not hold for the response times. We see that adding variability to the job size distribution also increases the mean response



(a) Erlang job sizes with mean one and SCV on the x-axis.

(b) Hyperexponential job sizes with balanced means, mean one and the SCV on the x-axis.

Fig. 12. Mean response time $\left(1 + \int_0^\infty \bar{F}(s)^d ds\right)$ as a function of the job sizes' SCV for $d = 2, 3, 4, 5$ and $\lambda = 0.45$.

time (for λ sufficiently bounded away from instability). From Figure 11b it is clear that only for small values of λ there is a reduction in response time by increasing d: this reduction is due to the fact that for small arrival rates a job is more likely to find an idle server by increasing d, but as λ increases higher values of d cause too much extra load on the servers which causes an increased response time. In all plots in Figure 11 we observe that mean response times stay relatively small until λ is close to λ_{max} at which point the mean response time explodes to infinity. This effect is more visible as the SCV of the jobs decreases and as the value of d increases.

6.2 Impact Of Job Size Variability

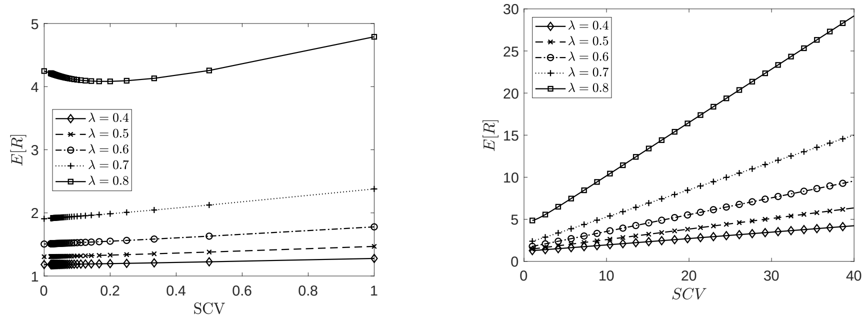
We now investigate how well Red_{eq}(d) behaves as a function of the job sizes distribution's SCV. In Figure 12 we show the mean response time as a function of the job sizes' SCV. On $[0, 1]$ we use deterministic, Erlang and exponential job sizes with mean one (see Figure 12a), while on $(1, 40]$ we use a hyperexponential distribution with balanced means and mean one (see Figure 12b). In both figures, we fix $\lambda = 0.45$ and $d = 2, 3, 4, 5$. We observe that the mean response time generally increases more or less linear as a function of the SCV. The most notable exception occurs when $d = 5$ and the SCV is close to zero. In this case $\mathbb{E}[R]$ decreases as the system with deterministic jobs is close to instability and increasing the job size variability somewhat increases the value of λ_{max}, which causes $E[R]$ to decrease. Moreover, we observe that increasing d in case of deterministic job sizes also increases the mean response time, whilst for large SCV this is not necessarily the case. This makes sense as for jobs with low variability, the risk of picking a server that is serving a large job is smaller than for more variable jobs, meaning there is less incentive to increase d and increasing d also increases the amount of work on the servers.

In Figure 13 we make the same plots as in Figure 12, but now instead of fixing λ and taking multiple values of d, we fix $d = 2$ and consider multiple values of λ. As expected, we observe that increasing λ also increases the mean response time and the slope of $\mathbb{E}[R]$ as a function of the SCV. Moreover we again observe that a decrease for low job size variability occurs when the system is close to instability.

6.3 Stability

Let $\bar{F}$ satisfy (9), we then find for all t for which $\bar{F}(s) \geq 0, s \in [0, t]$:

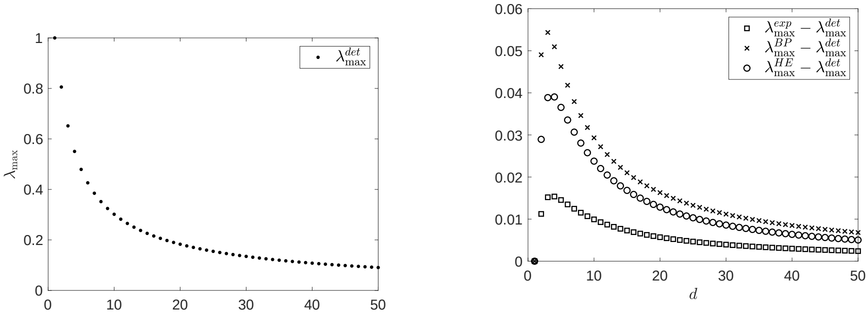
$$\bar{F}'(s) \leq -\lambda d \bar{G}(s)(1 - \bar{F}(s)).$$



(a) $d = 2$, $\lambda = 0.4, 0.5, 0.6, 0.7, 0.8$ and Erlang job sizes with mean one and SCV on the x-axis.

(b) $d = 2$, $\lambda = 0.4, 0.5, 0.6, 0.7, 0.8$ and hyperexponential job sizes with mean one and SCV on the x-axis.

Fig. 13. Mean response time $\left(1 + \int_0^\infty \bar{F}(s)^d ds\right)$ as a function of the job sizes' SCV.



(a) Deterministic job sizes.

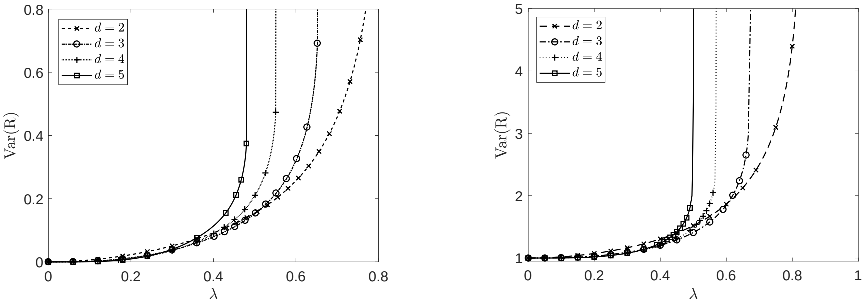
(b) The difference between exponential (exp), bounded Pareto (BP) and hyperexponential (HE) job sizes and deterministic (det) job sizes.

Fig. 14. The evolution of $\lambda_{\max}$ as a function of d for deterministic, exponential, bounded Pareto and hyperexponential job sizes in the Red$_{\text{eq}}$(d) model.

This shows that, if there exists an $\varepsilon > 0$ for which $\bar{G}(\varepsilon) > 0$ (i.e. job sizes are not identically zero) and $\bar{F}(0) < 1$, then in the limit $d \longrightarrow \infty$ we find that $\bar{F}(s)$ falls off at an unbounded speed. This shows that as d tends to infinity, $\lambda_{\max}$ tends to zero. It is also intuitively clear that this is the case as for $d = N$ we find that Red$_{\text{eq}}$(d) becomes an M/G/1 queue with arrival rate λN where N tends to infinity.

This fact is also reflected in Figure 14a, where we show the evolution of $\lambda_{\max}$ as a function of d for Red$_{\text{eq}}$(d) with deterministic job sizes. We observe that for $d = 1$, $\lambda_{\max} = 1$ (naturally as this is simply an M/D/1 queue), as d increases there is first a sharp drop in $\lambda_{\max}$ until $\lambda_{\max} \approx 0.2$ around $d = 20$ after which we see that the curve slowly converges to its horizontal asymptote at $\lambda_{\max} = 0$.

In Figure 14b, we compare the value of $\lambda_{\max}$ for other job sizes to $\lambda_{\max}$ for deterministic job sizes. We observe that the difference starts at zero (as for any M/G/1 queue $\lambda_{\max} = 1$), then jumps up for $d = 2$ and $d = 3$ after which it decays to zero. We observe that the difference in stability region increases as the tail of the job size distribution is more fat. The difference in $\lambda_{\max}$ is however fairly modest (no larger than 0.06).



(a) Deterministic job sizes.

(b) Exponential job sizes.

Fig. 15. The variance of the Response time distribution as a function of the arrival rate λ for $d = 2, 3, 4, 5$ in the Red$_{\text{eq}}$(d) model.

6.4 Variance of the Response Time Distribution

We now take a closer look at the behaviour of the variance of the response time distribution. To compute the variance, it is best to first compute the ccdf of the squared response time $\bar{F}_{R^2}$ and then integrate, i.e. we compute the variance as:

$$\text{Var}(R) = \int_0^\infty \bar{F}_{R^2}(s)\,ds - \left(\int_0^\infty \bar{F}_R(s)\,ds\right)^2.$$

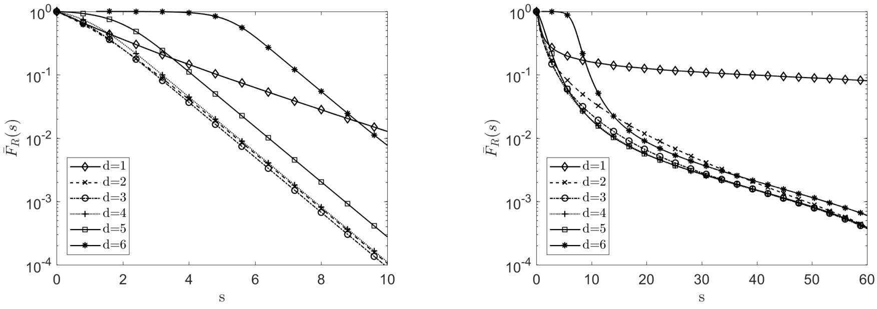
This is numerically more stable as it avoids the need to differentiate $\bar{F}_R$ to obtain its density f_R. Do note that the computation of $\bar{F}_{R^2}$ requires a quadratically wider s range $[0, s_{\max}]$ than $\bar{F}_R$ in order to ensure that $\bar{F}_{R^2}(s_{\max})$ is sufficiently small.

In Figure 15 we show the variance of the response time distribution as a function of the arrival rate λ for deterministic and exponential job sizes. We observe in Figure 15a that Var(R) remains very small until it explodes when λ approaches $\lambda_{\max}$. As long as $\lambda < \lambda_{\max} - 0.01$, we observe that Var($R$) < 1. When taking a closer look at the curves for small λ, we see that the variance decreases as d increases, this is due to the fact that, for low loads, having more replicas increases the chance of finding an empty server. However, as d increases, $\lambda_{\max}$ decreases which makes Var(R) explode to infinity faster.

For exponential job sizes (see Figure 15b), we also observe that the variance explodes as λ approaches $\lambda_{\max}$ and for small λ we still have a higher variance for smaller d. We see that for small values of λ the variance stays around 1 (this is due to the fact that for small loads all incoming jobs have a high probability of finding an idle server). The variance does however increase more quickly than for deterministic job sizes. Whereas for deterministic job sizes the workload at all queues increases at a similar speed, for an exponential job size distribution the workloads at the different servers will be less balanced as an arrival of a large job drives the workload of the d selected servers up by a large amount.

6.5 Tail of the Response Time distribution

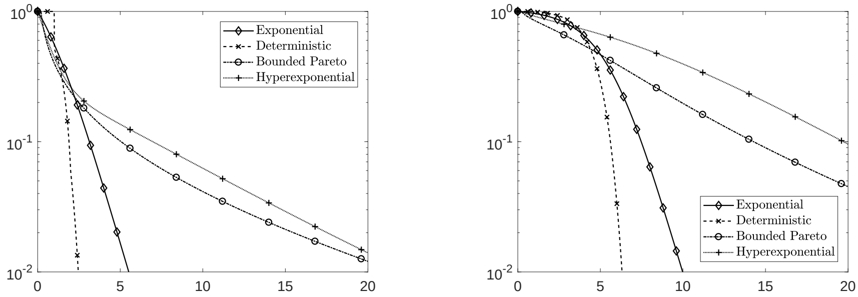
In Figure 16 we show $\bar{F}_R$ for $d = 1, \ldots, 6$ and $\lambda = 0.48$. We note that for both exponential and bounded Pareto job sizes, the system becomes unstable for $d \geq 7$, therefore these are no longer shown. We see in both Figures 16a and 16b that for identical replicas ($d \geq 2$) the tail of the response time distribution is lighter than for a classic M/G/1 queue ($d = 1$). However, the probability of having a very small response time is larger for the M/G/1 queue than for the case of identical



(a) Red$_{\text{eq}}$(d) with exponential job sizes.

(b) Red$_{\text{eq}}$(d) with bounded Pareto job sizes.

Fig. 16. Logarithmic plot of the response time distribution for Red$_{\text{eq}}$(d) with exponential resp. bounded Pareto job sizes, $\lambda = 0.48$ and varying values of d.



(a) Red$_{\text{eq}}$(d) for varying job sized and $\lambda = 0.5$.

(b) Red$_{\text{eq}}$(d) for varying job sized and $\lambda = 0.8$.

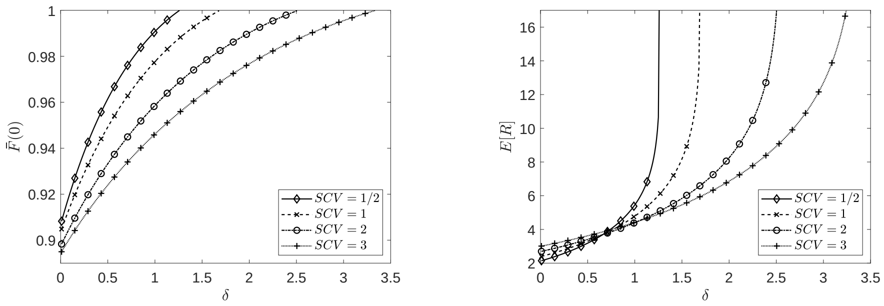
Fig. 17. Logarithmic plot of the response time distribution for Red$_{\text{eq}}$(d) for all main job size distributions, $\lambda = 0.5$ resp. $\lambda = 0.8$ and $d = 2$.

replicas, especially for $d = 6$. This is due to the fact that, while replicas decrease the probability that a job ends up in a long queue, the queues are more heavily loaded, which decreases the probability of finding a queue with a very small workload (especially for $d = 6$ as the system is close to instability in this case). This figure also confirms that as the job size variability increases, the gain from having replicas increases.

In Figure 17a we compare the ccdf for various job size distributions, $d = 2$ and $\lambda = 0.5$. The tails behave as expected: the fatter the tail of the job size distribution, the fatter the tail of the response time distribution. However what is interesting to note is that for sufficiently small values of s (around $s < 4$), the value of $\bar{F}(s)$ is greater for the less variable job sizes. Moreover, when looking at Figure 17b, we see that this effect is strengthened when λ increases. This can be understood by recalling that λ_{max} is smaller for less variable job size distributions.

6.6 Red$_{\text{eq}}$(d) with cancellation delay

In this subsection, we look at the impact of having a cancellation delay on the performance of Red$_{\text{eq}}$(d). In Figure 18 we take $\lambda = 0.7$, $d = 2$ and consider 4 distributions for X: Erlang with 2 phases, mean one and SCV $1/2$, Exponential and Hyperexponential with balanced means and $SCV = 2$ and 3. We observe in Figure 18a that the system load increases in a concave manner as the cancellation



(a) Plot of the system load as a function of the cancellation delay.

(b) Plot of the mean response time as a function of the cancellation delay.

Fig. 18. Plots for Red$_{eq}$(d) with cancellation delay for X Erlang with 2 phases, mean one and SCV 1/2, Exponential and Hyperexponential with balanced means and $SCV = 2, 3$, $d = 2$ and $\lambda = 0.7$.

delay δ increases. Moreover we observe that as job sizes are more variable, this increase is less steep. Further, we observe in Figure 18b, that while for a small delay, the mean response time increases as the job size variability increases, this relation is reversed for a delay of $\delta \geq 0.8$. Finally we note that the system becomes unstable as we increase the cancellation delay δ, and the maximum point of stability $\lambda_{\max}$ increases as the SCV of the job size increases.

7 COMPARISON RED$_{\mathbf{eq}}$(D) AND RED$_{\mathbf{iid}}$(D)

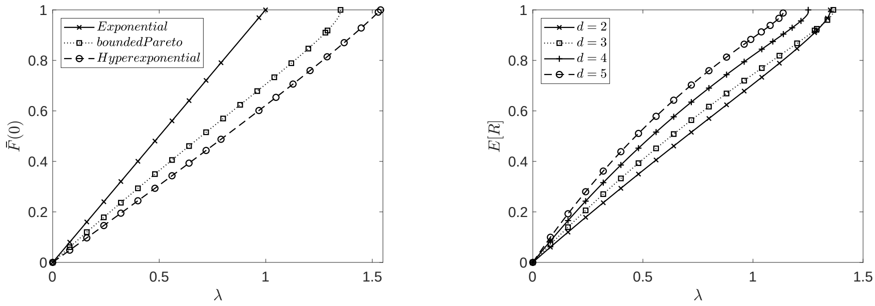
In this section, we take a look at Red$_{iid}$(d) by comparing it to Red$_{eq}$(d). We first show that Red$_{iid}$(d) always performs better than Red$_{eq}$(d). Afterwards, we revisit some of the numerical experiments from Section 6 to indicate key differences between the two models. These results show why it may be misleading to apply Red$_{iid}$(d) as a model when the replica sizes are in fact not independent.

7.1 Red$_{\mathbf{iid}}$(d) stochastically outperforms Red$_{\mathbf{eq}}$(d)

Through a simple coupling argument we can show that the workload (and thus also the response time) for Red$_{iid}$(d) is always lower than for Red$_{eq}$(d).

Proposition 7.1. *Suppose we have $d \leq N < \infty$ servers, incoming job sizes follow a general distribution and the arrival process has a general distribution. Then the workload and response time distribution of the system of N servers operating under Red$_{iid}$(d) is always stochastically lower than for the same system operating under Red$_{eq}$(d).*

Proof. Suppose the two systems are coupled such that arrivals occur at the same time and choose the same servers in both systems. At the first arrival, both systems are still empty and the new workload of a selected server for Red$_{iid}$(d) is given by $\min_{i=1}^{d} S_i$ and S for Red$_{eq}$(d), where $S, S_1, \ldots, S_d$ are independent and have distribution G. The inequality $\min_{i=1}^{d} S_i \leq_d S$ obviously holds (where $\leq_d$ denotes the distributional inequality). By induction, we assume that at an arbitrary arrival instant in the future, the chosen servers for Red$_{iid}$(d) have workload $U_1, \ldots, U_d$ while the chosen servers for Red$_{eq}$(d) have workload $V_1, \ldots, V_d$ with $U_i \leq_d V_i$ for all i. We find (with



(a) $d = 2$ and different job size distributions.

(b) $d = 2, 3, 4, 5$ and bounded Pareto job sizes.

Fig. 19. Workload $\bar{F}(0)$ as a function of the arrival rate λ for Red$_{\text{iid}}$(d). This Figure should be compared to Figure 10.

$S, S_1, \dots, S_d$ independent random variables with distribution G):

$$\max\left\{U_1, \min_{i=1}^{d}\{U_i + S_i\}\right\} \leq_d \max\left\{V_1, \min_{i=1}^{d}\{V_i + S_i\}\right\} \leq_d \max\left\{V_1, \min_{i=1}^{d}\{V_i\} + S\right\}$$

this shows (by permuting $U_1, \dots, U_d$) that after an arrival instant the workload for Red$_{\text{iid}}$(d) is still stochastically smaller than that of Red$_{\text{eq}}$(d). As the service is constant at rate 1 in both systems this shows that the inequality regarding the workload process indeed holds.

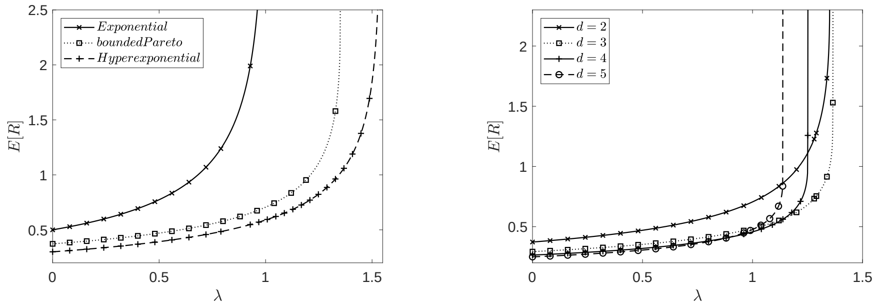
For the response times, we note that when a job as the one above arrives, its response time is exactly given by $\min_{i=1}^{d}\{U_i + S_i\}$ resp. $\min_{i=1}^{d}\{V_i\} + S$ for the Red$_{\text{iid}}$(d) resp. Red$_{\text{eq}}$(d) model. This shows by the above discussion that the distributional inequality for the response time distributions also holds. □

Remark. *For deterministic job sizes equal to one, it is obvious that Red$_{\text{eq}}$(d) and Red$_{\text{iid}}$(d) become equivalent. We can also show this analytically by looking at Theorem 4.2 and Proposition 4.6, in Proposition 4.6 we find for deterministic job sizes that: $\bar{F}_{R_1}(s) = 1, s \leq 1$ and $\bar{F}_{R_1}(s) = \bar{F}(s-1)$ otherwise. This allows one to verify that (17) reduces to the same expression as (9) in case of deterministic job sizes.*

7.2 Mean Response Time and Workload distribution

In this section, we take another look at the setting in Section 6.1 for Red$_{\text{iid}}$(d) instead of Red$_{\text{eq}}$(d). Figure 19 shows the workload as a function of the arrival rate λ and should be compared to Figure 10. We observe in Figure 19a that the workload equals λ for exponential job sizes (a fact which is shown in [6]). For bounded Pareto and hyperexponential job sizes, we also observe a close to linear growth as a function of λ with a less steep slope, implying that the stability region is larger for the more variable job size distributions. This is to be expected as the minimum of two more variable job size distributions has a smaller mean than the minimum of two exponential distributions. In Figure 19b we observe that, despite the fact that workload decreased when going from $d = 1$ to $d = 2$, the workload increases when we further increase the value of d in case of bounded Pareto job sizes. This is further discussed in Section 7.4, where the stability is investigated.

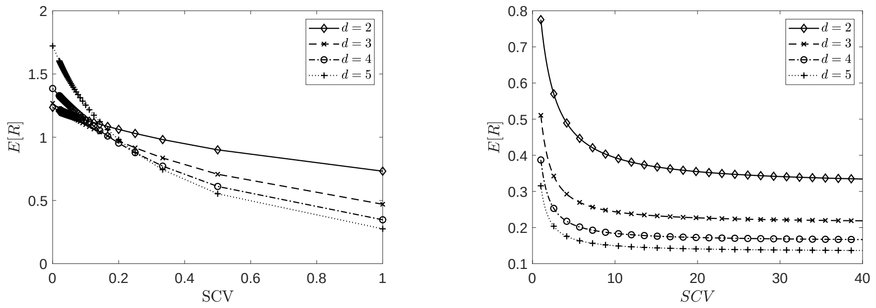
In Figure 20, we observe the mean response time as a function of λ for the same settings as in Figure 19 (it should be compared to Figure 11). We observe some similarity between Red$_{\text{eq}}$(d) and Red$_{\text{iid}}$(d): the mean response time is very low until λ gets very close to $\lambda_{\max}$ at which point it snaps



(a) $d = 2$ and different job size distributions.

(b) $d = 2, 3, 4, 5$ and bounded Pareto job sizes.

Fig. 20. Mean response time $\left(1 + \int_0^\infty \bar{F}(s)^d ds\right)$ as a function of the arrival rate λ for Red$_{\text{iid}}$(d). This Figure should be compared to Figure 11.



(a) Erlang job sizes with mean one and SCV on the x-axis.

(b) Hyperexponential job sizes with balanced means, mean one and SCV on the x-axis.

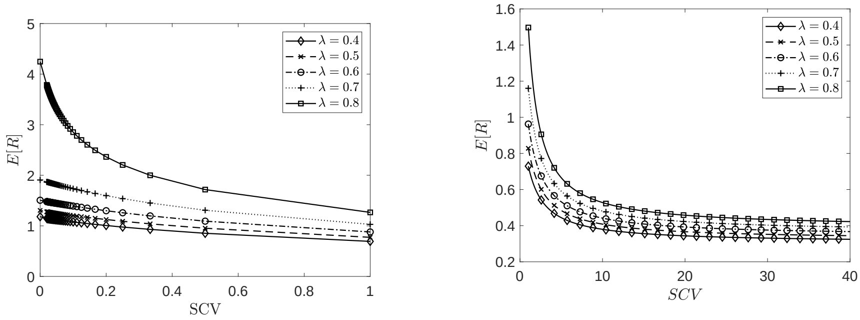
Fig. 21. Mean response time $\left(1 + \int_0^\infty \bar{F}(s)^d ds\right)$ as a function of the job sizes' SCV for $d = 2, 3, 4, 5$ and $\lambda = 0.45$ for the Red$_{\text{iid}}$(d) model. This Figure should be compared to Figure 12.

and goes to infinity. It is obvious that mean response times for Red$_{\text{iid}}$(d) lies far below the mean response times for Red$_{\text{eq}}$(d) and also the point at which it snaps (i.e. $\lambda_{\max}$) lies way further to the right.

7.3 Impact Of Job Size Variability

In this section, we take the same setting as in Section 6.2. We observe in Figure 21 that Red$_{\text{iid}}$(d) behaves completely different as a function of the SCV compared to Red$_{\text{eq}}$(d). The mean response time decreases sharply as the SCV increases, moreover taking a higher value of d is always beneficial irrespective of the job size variability. In Figure 22, we observe that increasing the value of λ has no effect on the behaviour of $\mathbb{E}[R]$ w.r.t. the SCV, it still decreases monotonically as the SCV increases. These Figures should be compared to Figures 12 and 13. These results further illustrate the inappropriateness of assuming independence for systems where the replicas do not have independent sizes.

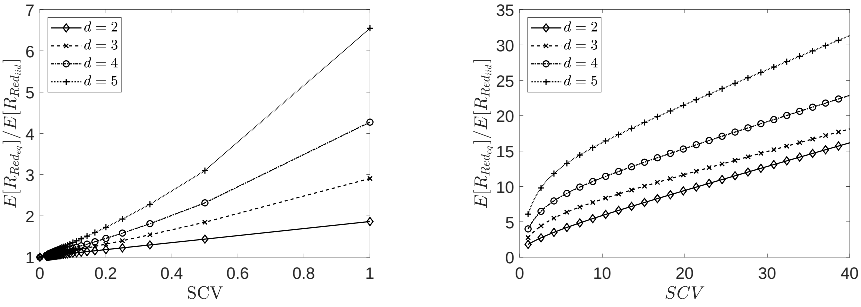
Things are even more clear when we take the quotient of the mean response time for the Red$_{\text{eq}}$(d) policy and the Red$_{\text{iid}}$(d) policy for the same parameter settings. In Figure 23, we show the quotient



(a) $d = 2$, $\lambda = 0.4, 0.5, 0.6, 0.7, 0.8$ and Erlang job sizes with mean one and SCV on the x-axis.

(b) $d = 2$, $\lambda = 0.4, 0.5, 0.6, 0.7, 0.8$ and hyperexponential job sizes with mean one and SCV on the x-axis.

Fig. 22. Mean response time $\left(1 + \int_0^\infty \bar{F}(s)^d ds\right)$ as a function of the job sizes' SCV for the Red_{iid}(d) model. This Figure should be compared to Figure 13.



(a) Erlang job sizes with mean one and SCV on the x-axis.

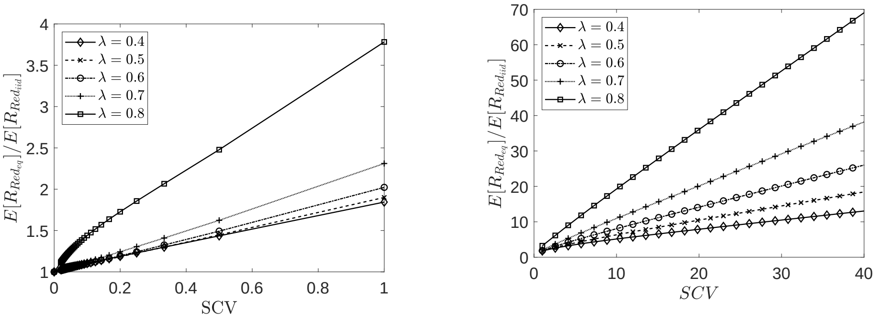
(b) Hyperexponential job sizes with balanced means, mean one and the SCV on the x-axis.

Fig. 23. Quotient of the mean response time for identical replicas and independent replicas as a function of the job sizes' SCV for $d = 2, 3, 4, 5$ and $\lambda = 0.45$.

of the data found in Figure 12 and Figure 21. We observe that increasing the job size variability and the number of chosen servers d both increase the mismatch between Red_{eq}(d) and Red_{iid}(d) dramatically. Furthermore Figure 24 depicts the quotient of the mean response times for Red_{eq}(d) resp. Red_{iid}(d) found in Figures 13 resp. Figure 22. We observe that the mismatch is even further increased by taking a higher value for λ.

7.4 Stability

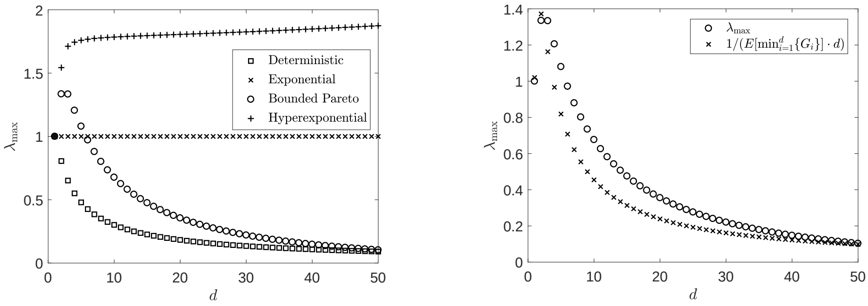
We now reuse the setting in Figure 14, the results are shown in Figure 25a. For deterministic job sizes the value of $\lambda_{\max}$ is obviously identical for Red_{iid}(d) and Red_{eq}(d). For other distributions, we observe a completely different picture, for hyperexponential job sizes: $\lambda_{\max}$ increases quickly at first and then increases to a horizontal asymptote. For exponential job sizes the value of $\lambda_{\max}$ is constant and equal to 1 (as shown in [6]). For bounded Pareto job sizes, we observe that first, $\lambda_{\max}$ increases but afterwards it decreases to zero. It is in fact easy to show that for any job size distribution which has a lower bound, the value of $\lambda_{\max}$ decreases to zero. Indeed, if the job sizes



(a) $d = 2$, $\lambda = 0.4, 0.5, 0.6, 0.7, 0.8$ and Erlang job sizes with mean one and SCV on the x-axis.

(b) $d = 2$, $\lambda = 0.4, 0.5, 0.6, 0.7, 0.8$ and hyperexponential job sizes with mean one and SCV on the x-axis.

Fig. 24. Quotient of the mean response time for identical replicas and independent replicas as a function of the job sizes' SCV for the Red$_{\text{iid}}$(d) model.



(a) Different job size distributions. This Figure should be compared to Figure 14

(b) Plot of how well $1/(\mathbb{E}[\min_{i=1}^{d} G_i] \cdot d)$ approximates $\lambda_{\max}$ for bounded Pareto job sizes.

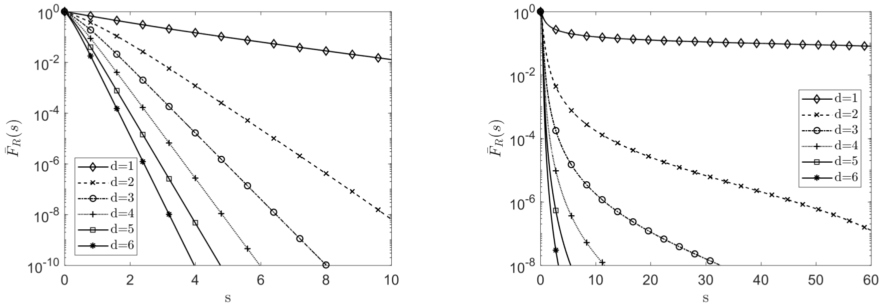
Fig. 25. Plot of $\lambda_{\max}$ as a function of d for Red$_{\text{iid}}$(d) .

are lower bounded by a value $a > 0$, then we find for all $s < a$ that $\bar{F}_{R_1}(s) = 1$, therefore by (17) $\bar{F}'(s) = -\lambda d(1 - \bar{F}(s))$ for $s < a$. This ODE has the solution $\bar{F}(s) = 1 - (1 - \bar{F}_0)e^{\lambda d s}$ (with $\bar{F}(0) = \bar{F}_0$). As $d \longrightarrow \infty$ we see that for any $\bar{F}_0 < 1$, $\bar{F}(s)$ decreases to $-\infty$ and is thus not a ccdf.

For hyperexponential job sizes, it seems like $\lambda_{\max}$ converges to some constant around 1.8. We can indeed show that this is the case: the value of $\lambda_{\max}$ for sufficiently large d is approximated by $1/(\mathbb{E}[\min_{i=1}^{d}\{G_i\}] \cdot d)$, where G_i are i.i.d. distributed as G. For $d = N$ this approximation is exact for any job size distribution as the queue behaves like an M/G/1 queue with arrival rate λN and processing time $\min_{i=1}^{d}\{G_i\}$. For exponential job sizes we have $\mathbb{E}[\min_{i=1}^{d}\{G_i\}] \cdot d = 1$ for all d. For hyperexponential job sizes we find $\lim_{d\to\infty} \lambda_{\max} = 1.79\ldots$. We illustrate this approximation as a function of d for bounded Pareto job sizes in Figure 25b.

7.5 Tail of the Response Time distribution

In Figure 26 we show the tail of the response time distribution when replicas are independent (for the same setting as when replicas were assumed to be identical in Figure 16). For independent replicas the discussion is much simpler, as both the exponential and bounded Pareto distribution are



(a) Red_{iid}(d) with exponential job sizes.

(b) Red_{iid}(d)with bounded Pareto job sizes.

Fig. 26. Logarithmic plot of the response time distribution for Red_{iid}(d) for exponential resp. bounded Pareto job sizes with $\lambda = 0.48$ and varying values of d. These Figures shoul be compared to Figure 16.

sufficiently variable, there is a clear advantage by making independent replicas. When considering distributions with a lighter tail, the results would be more similar to the case of identical replicas (see e.g. Figure 22a versus Figure 13a).

8 FUTURE WORK

An important generalization is to look at the $S\&X$ model introduced in [5]. The Red_{eq}(d) model corresponds to the $S\&X$ model with no slowdown (i.e., $S = 1$), which implies that the replica that starts execution first also finishes first. As such it is always better to cancel the other replicas as soon as one starts execution. The Red_{iid}(d) model corresponds to the $S\&X$ model with deterministic job sizes (i.e., $X = 1$), which implies that if job sizes are more variable than exponential, extra replicas reduce latency. As such it is always better to replicate on as many servers as possible. However, with the $S\&X$ model different replicas may experience different slowdowns and cancellation-on-start may no longer be superior. It is not hard to obtain general expressions for $c_d(t, s, r)$ and $C_d(t, r)$ for the $S\&X$ model, which may lead to a similar differential equation with unknown boundary condition. Proving Conjecture 3.3 would give a theoretical basis for the analysis provided here (as was done for other load balancing schemes in [4]). We note that this is also an open problem for replication with i.d.d. replicas considered in [6]. It might be worth trying to explicitly solve the DIDE (9) for certain job size distributions.

REFERENCES

[1] G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica. 2013. Effective Straggler Mitigation: Attack of the Clones. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation (nsdi'13)*. USENIX Association, Berkeley, CA, USA, 185–198. http://dl.acm.org/citation.cfm?id=2482626.2482645

[2] Urtzi Ayesta, Tejas Bodas, and Ina Maria Verloop. 2018. On a unifying product form framework for redundancy models. *Performance Evaluation* 127 (2018), 93–119.

[3] 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

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

[5] 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 Trans. Netw.* 25, 6 (Dec. 2017), 3353–3367. https://doi.org/10.1109/TNET.2017.2744607

[6] 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. https://doi.org/10.1287/opre.2016.1582

[7] T. Hellemans and B. Van Houdt. 2018. On the Power-of-d-choices with Least Loaded Server Selection. *Proc. ACM Meas. Anal. Comput. Syst.* 2 (2018), Article No. 27. Issue 2.

[8] Y. Raaijmakers, S. Borst, and O. Boxma. 2018. Delta probing policies for redundancy. *Performance Evaluation* 127-128 (2018), 21 – 35. https://doi.org/10.1016/j.peva.2018.09.002

[9] Y. Raaijmakers, S. Borst, and O. Boxma. 2018. Redundancy scheduling with scaled Bernoulli service requirements. *arXiv preprint* (2018). http://arxiv.org/abs/1811.06309 arXiv:1811.06309.

[10] Z. Schuss. 2009. *Theory and applications of stochastic processes: an analytical approach*. Vol. 170. Springer Science & Business Media.

[11] N. B. Shah, K. Lee, and K. Ramchandran. 2016. When Do Redundant Requests Reduce Latency? *IEEE Transactions on Communications* 64, 2 (Feb 2016), 715–722. https://doi.org/10.1109/TCOMM.2015.2506161