Efficient scheduling in redundancy systems with general service times

Efficient scheduling in redundancy systems with general service times

Elene Anton[1], Rhonda Righter[2] and Ina Maria Verloop[3,4]

[1] Eindhoven University of Technology TU/e, Eindhoven, The Netherlands
[2] University of California at Berkeley, Berkeley, USA
[3] CNRS, IRIT, Toulouse, France
[4] Université de Toulouse INP, Toulouse, France

Abstract

We characterize the impact of scheduling policies on the mean response time in nested systems with cancel-on-complete redundancy. We consider not only redundancy-oblivious policies, such as FCFS and ROS, but also redundancy-aware policies of the form $\Pi_1 - \Pi_2$, where Π_1 discriminates among job classes (e.g., least-redundant-first (LRF), most-redundant-first (MRF)) and Π_2 discriminates among jobs of the same class. Assuming that jobs have independent and identically distributed (i.i.d.) copies, we prove the following: (i) When jobs have exponential service times, LRF policies outperform any other policy. (ii) When service times are New-Worse-than-Used, MRF-FCFS outperforms LRF-FCFS as the variability of the service time grows infinitely large. (iii) When service times are New-Better-than-Used, LRF-ROS (resp. MRF-ROS) outperforms LRF-FCFS (resp. MRF-FCFS) in a two-server system. Statement (iii) also holds when job sizes follow a general distribution and have identical copies (all the copies of a job have the same size). Moreover, we show via simulation that, for a large class of redundancy systems, redundancy-aware policies can considerably improve the mean response time compared to redundancy-oblivious policies. We also explore the effect of redundancy on the stability region.

Key words: scheduling, redundancy, performance.

1 Introduction

In the present paper we investigate the impact that the scheduling policy has on the performance of redundancy systems when the usual exponentially distributed i.i.d. copies assumption is relaxed. In particular, we investigate the performance, in terms of the total number of jobs in the system, for two classes of scheduling policies: redundancy-oblivious policies and redundancy-aware policies. Redundancy-unaware policies are for instance FCFS (First-Come-First-Serve) and ROS (Random-Order-of-Service), where the server is oblivious to the job class. In contrast, under redundancy-aware policies, the scheduler tries to exploit the knowledge of the distribution of the redundant copies in the system. We consider redundancy-aware policies that are composed of two levels. The first level (Π_1) describes the priority among the job classes and the second level (Π_2) describes how the jobs with the same priority are served in a server. Examples of first-level policies are LRF (Least-Redundant-First) and MRF (Most-Redundant-First), where under LRF, respectively MRF, within a server jobs with fewer copies, respectively more copies, have priority over jobs with more copies, respectively fewer copies. Second-level policies could be FCFS or ROS. These policies do not depend on the system state, so are easily implementable and scalable.

The main motivation for studying the impact of redundancy in multi-server systems comes from the fact that both empirical ([3, 4, 10, 26]) and theoretical ([7, 12, 16, 18, 21, 25]) evidence shows that redundancy can improve the performance in real-world applications. Under redundancy, an arriving job dispatches multiple copies to all compatible servers, and departs either when a first copy enters service (known as the cancel on start, $c.o.s.$, model) or when a first copy completes service (known as the cancel on complete, $c.o.c.$ model). We focus on c.o.c. models. Redundancy aims to exploit the variability of the queue lengths and server capacities, potentially reducing the response time. However, adding redundant copies also may waste resources in the additional servers that do not complete the copy. Hence, the potential of redundancy relies on finding scheduling policies that improve the latency of jobs while not overloading the system.

Stability of redundancy models has been studied in recent work for exponential service times. Anton et al. [5] provide an overview of the stability results for redundancy models. Under the FCFS scheduling policy and when jobs have independent and identically distributed (i.i.d.) copies, Gardner et al. [13, 16] and Bonald and Comte [8] fully characterize the stability region and show that it is not reduced due to adding redundant copies. Furthermore, the authors show that the stationary distribution of this model is of a product-form. More precisely, this model falls in the framework of more general systems described in Gardner and Righter [15] that present a product-form steady-state distribution. Anton et al. [7] show that the latter stability result also holds for the redundancy-d model when either PS or ROS is implemented in the servers.

Motivated by the evidence in Vulimiri et al. [26] that the i.i.d. copies assumption might be unrealistic, Anton et al. [6, 7] assume that copies are identical, that is, all the copies of a job have the same size as the original job. For exponential service times, the authors observe that the stability condition strongly depends on the scheduling policy implemented in the servers. In particular, the stability region of the redundancy-d model is not reduced when the scheduling policy is ROS, but it is dramatically reduced when the scheduling policy is either FCFS or PS.

Raaijmakers et al. [23] relax the exponential service time assumption to consider New-Better-than-Used (NBU) or New-Worst-than-Used (NWU) service times. They study the redundancy-d model under FCFS, where jobs have identical copies, server capacities are heterogeneous and all servers sample independent speed variations for each copy in service from a general distribution. The authors show that under NBU service distributions, the stability region under $d = 1$ is larger than that under $d > 1$. When the service time distribution is instead NWU, the authors observe that the stability condition strongly depends on the load in the system.

The impact of the redundancy policy on the number of jobs in the system was first studied in [19, 20] for the FCFS scheduling policy, where each job can dispatch i.i.d. copies to any server in the system. Assuming NWU service time distributions, Koole and Righter [20] show that full replication stochastically minimizes the number of jobs in the system at any time. In contrast, for NBU service time distributions, Kim et al. [19] show that no-replication is optimal.

In [1, 2, 11, 14], redundancy-aware policies are introduced. Akgun et al. [1, 2] consider a $c.o.s.$ system where each server has dedicated traffic, that is, each server receives jobs of a class that does not send copies to other servers. The authors consider the DCF (Dedicated-Customers-First) service policy and analyze the efficiency and fairness for both dedicated and redundant jobs. Gardner et al. [11, 14] investigate the impact that the implemented scheduling policy has on the performance for nested $c.o.c.$ redundancy models with exponential service times and i.i.d. copies. The authors introduce Least-Redundant-First (LRF) and Primaries-First (PF) scheduling policies. Under PF, each job has a copy that is a primary copy and the rest of the copies are secondary copies. Within each server, primary copies have priority over secondary copies, and within a priority level, copies are served in order of arrival. In [11], the authors consider the W-model and observe that implementing FCFS in the servers is highly effective in reducing the mean response time in the system, even though LRF is optimal. However, LRF fails to be fair to non-redundant

jobs. Thus, the authors propose PF (Primaries-First), which minimizes the overall mean response time, subject to a fairness condition. In Gardner et al. [14], the authors consider general nested systems and show that for LRF, even if scheduling more redundant jobs is better, the maximum gains come from adding only a small proportion of redundant jobs.

Nageswaran et al. [22] consider the N-model under FCFS and exponentially distributed job sizes and i.i.d. copies where the redundant jobs are scheduled either under $c.o.c.$ or $c.o.s.$. The authors analyze the mean response time per class and characterize under which conditions redundancy is fair compared to the JSQ system without redundancy.

In this paper, we compare the performance under different redundancy-aware policies for general service time distributions and for both i.i.d. and identical copies. We introduce two-level priority policies Π_1-Π_2, where the first level policy Π_1 determines the priorities among the job classes, and the second-level policy Π_2 determines the policy among the jobs with the same priority. Below we describe our main contributions, which are summarized in Table 1.

Under the i.i.d. copies assumption, we show that for the nested redundancy model with exponential service times, LRF-Π_2 minimizes the number of jobs in the system independently of the non-idling second-level policy Π_2. That is, we generalize the result in [11] to any non-idling second-level policy Π_2. Furthermore, when service times are NWU, we show that for a given non-idling first-level policy Π_1, Π_1-FCFS minimizes the number of jobs in the system. The intuition for the latter comes from the fact that under NWU service times and i.i.d. copies, the service time of a copy that enters service is stochastically smaller than the remaining service time of a copy of that job that is already in service on another server, which increases the chance that the job departs sooner. We further prove that the optimal first-level policy under NWU service times depends on the variability in the service times. In particular, we show that as the coefficient of variation grows large, MRF becomes optimal, while LRF is optimal when the coefficient of variation is one (exponential service times).

In the case of NBU service times, only partial characterizations of an optimal policy are de-

i.i.d. copies			
	Service times:		
	Exponential	**NWU distributions**	
Model:	**Nested**	**Nested**	**General**
Result:	LRF-$\Pi_2 \succ \pi$ (Prop. 2)	MRF-FCFS $\succ$ LRF-FCFS as $q \to 0$ (Prop. 4), where the variability of the service time increases as $q \to 0$	Π_1-FCFS $\succ$ Π_1-Π_2 (Prop. 3)
		NBU distributions	
Model:		**Nested**	**W-model**
Result:		LRF-FCFS $\succ$ MRF-FCFS, with λ small enough, deterministic (Prop. 7)	Π_1-ROS $\succ$ Π_1-FCFS (Corol. 6)

identical copies and general service times		
Model:	**Nested**	**W-model**
Result:	MRF-$\Pi_2 \succ$ MRF-FCFS (Prop 8)	Π_1-ROS $\succ$ Π_1-FCFS (Corol. 10)
	LRF-FCFS $\succ$ MRF-FCFS, with λ small enough (Prop. 7)	

Table 1: Summary of policy comparison results. We write $\pi \succ \pi'$, if policy π has a better performance than policy π' with respect to the particular performance measure.

rived for two servers. When a server is non-idling, we show that for the second-level policy it is better to choose according to ROS than according to FCFS. This is intuitively clear, since the service time of a copy that is already in service is stochastically smaller than a copy that enters service when service times are NBU.

Under the identical copies assumption, we show that LRF is the best first-level policy when the arrival rate is small enough. In addition, we prove that for the nested redundancy model, MRF-Π_2 outperforms MRF-FCFS for any service time distributions, with Π_2 non-idling. The latter follows intuitively from the fact that when copies are identical, all the copies of each job have the same size, which necessarily induces a waste of resources when serving copies of the same job. For the W-model, our results are similar to those obtained for NBU distributions with i.i.d. copies. This similarity is explained by the fact that having identical copies means that a copy in service has always a smaller remaining service time than a copy of this job that is not yet in service, just as in the NBU i.i.d. case.

We also compared the performance when one can choose between i.i.d. copies and identical copies for a fixed policy (Section 3.2). In particular, we show that for a general redundancy model with general service times and for a given policy Π_1-FCFS, the total number of jobs when copies are i.i.d. is stochastically smaller than when copies are identical.

We also investigate the stability condition for the redundancy-aware scheduling policies analyzed in the this paper. In particular, for the redundancy system with a general topology and heterogeneous server capacities and exponentially distributed service times, we show that (i) for LRF-Π_2 with i.i.d. copies and non-idling Π_2, and (ii) for LRF-ROS with any correlation structure among the copies, the stability region is not reduced due to adding redundant copies.

Finally, we numerically compare redundancy-aware policies with redundancy-oblivious policies. We observe that when service variability is high and copies are i.i.d. or when copies are identical, it may be advantageous to use redundancy-aware policies.

2 Model description

We consider a K parallel server system with heterogeneous capacities μ_s, for $s \in S$, where $S = \{1, \dots, K\}$ is the set of all servers. Jobs arrive to the system according to a Poisson process of rate λ. Each job is labelled with a class c that represents the subset of servers to which it sends a copy: i.e., $c = \{s_1, \dots, s_n\} \subset S$, for some n. We denote by $\mathcal{C}$ the set of all classes in the system. An arriving job is with probability p_c of class c, with $\sum_{c\in\mathcal{C}} p_c = 1$. Let us denote by $\mathcal{C}(s) = \{c \in \mathcal{C} \ : \ s \in c\}$ the subset of classes that dispatch a copy to server s. Therefore, an arrival sends a copy to server $s \in S$ with probability $\sum_{c\in\mathcal{C}(s)} p_c$. We assume that the correlation structure among the copies is either i.i.d. or identical copies. We also assume that all copies of a job are canceled once any copy of this job completes.

In this paper, special attention will be given to the class of nested redundancy models. We call a redundancy model nested if the set of classes $\mathcal{C}$ satisfies the following: for all job classes $c, c' \in \mathcal{C}$, either *i)* $c \subseteq c'$ or *ii)* $c' \subseteq c$ or *iii)* $c \cap c' = \emptyset$. The smallest nested system is the so-called N-model: this is a $K = 2$ server system with classes $\mathcal{C} = \{\{2\}, \{1, 2\}\}$. Another nested system is the W-model, that is, $K = 2$ servers and classes $\mathcal{C} = \{\{1\}, \{2\}, \{1, 2\}\}$. In Figure 1, we illustrate the N-model, the W-model and a general nested model with $K = 4$. Another well-studied model is the redundancy-d model where each incoming job sends a copy to d out of K servers chosen uniformly at random. That is, $\mathcal{C} := \{\{s_1, \dots, s_d\} \subset S \ : \ s_i \neq s_j, \quad \forall i \neq j\}$, with $|\mathcal{C}| = \binom{K}{d}$ and $p_c = 1/\binom{K}{d}$. We refer to Figure 1 for an illustration of a redundancy-d model with $K = 4$ and $d = 2$. Nested models arise naturally in systems with data locality constraints, or with hierarchical dispatching.

We denote by π a generic scheduling policy implemented in the system. We assume that the

policy π has no information on the actual size of the copies, that is, is non-anticipating. It may depend on the system state. In this paper, we introduce two-level redundancy-aware scheduling policies denoted by $\pi = \Pi_1$-Π_2:

- The first-level policy Π_1 determines the preemptive priority among job classes. We assume the priority policy to be strict, that is, in each server s there is a strict priority ranking of all job classes in $\mathcal{C}(s)$.
- The second-level policy Π_2 determines the scheduling policy of jobs within the same class. This policy is assumed to be size-unaware and non-preemptive within the class. That is, once a job of a given class is started at a server, no other job of the same class can be served at that server until the given job has completed (at some server).

Examples of first-level policies Π_1 are Least-Redundant-First (LRF) and Most-Redundant-First (MRF). Note that these policies are uniquely defined for nested systems. Examples of second-level policies Π_2 are FCFS, LCFS, and ROS. These are also examples of single-level redundancy-oblivious policies.

For a given scheduling policy π, we denote by $N_c^\pi(t)$ the number of class-c jobs present in the system at time t and by $N^\pi(t) := \sum_{c\in\mathcal{C}} N_c^\pi(t)$ the total number of jobs in the system. We aim to compare the performance of the system, in terms of the number of jobs, under different scheduling policies. We have the following stochastic ordering definition.

Definition 1. *For two nonnegative continuous random variables X and Y, with respective distributions F and G, and $\bar{F}(x) = 1-F(x)$ and $\bar{G}(x) = 1-G(x)$, we say that $X \geq_{st} Y$, that is, X is stochastically larger than Y, if $E[h(X)] \geq E[h(Y)]$ for all increasing functions h. Equivalently, if $\bar{F}(x) \geq \bar{G}(x)$ for all $x \geq 0$.*

We let X denote the service time distribution of a job when it is served at capacity 1. Special focus will be given to exponential service times, as well as the following two classes of service time distributions: New-Worst-than-Used (NWU) and New-Better-than-Used (NBU), defined below. Let $X_t = [X - t \,:\, X > t]$ be the remaining processing time of a job that has completed t time units of service.

Definition 2. *We say that X is New Worse than Used (NWU) if the remaining processing time of a task that has received some processing (is used) is stochastically* larger *than the processing time of a task that has received no processing (is new), i.e., $X_0 \leq_{st} X_t$ for all t. We say that X is New Better than Used (NBU) if the remaining processing time of a task that has received some processing (is used) is stochastically* smaller *than the processing time of a task that has received no processing (is new), i.e., $X_t \leq_{st} X_0$ for all t.*

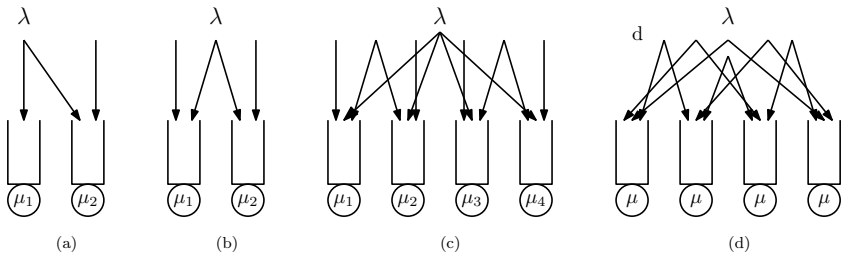


Figure 1: (a) the N-model, (b) the W-model, (c) a general nested model ($K = 4$), and (d) the redundancy-d model ($K = 4$ and $d = 2$).

A sufficient condition for X to be NWU (NBU) is for it to have decreasing (increasing) failure rate, that is, $h(x) = f(x)/\bar{F}(x)$ is decreasing (increasing) in x, where $f(x)$ and $F(x)$ are the probability density function and the cumulative distribution function of X, respectively. Additionally, if X has a decreasing (increasing) failure rate, then the coefficient of variation of X is at least (at most) 1. We note that the exponential service time distribution has constant failure rate, so that it belongs to both categories, NWU and NBU.

Throughout this paper, we provide numerical examples of the performance under the different policies. These numerics are obtained by Matlab where we run a large number of busy periods (10^6), so that the variance and confidence intervals of the mean number of jobs in the system are sufficiently small. The service time distributions that we consider are exponential, deterministic, and Weibull ($\bar{F}(x) = e^{(-x/\kappa)^\alpha}$, with $\kappa, \alpha > 0$). We note that the deterministic and Weibull distribution with $\alpha \geq 1$ are NBU distributions, while the Weibull distribution with $\alpha \leq 1$ is NWU. We also consider a class of degenerate distributions, where X is equal to Y/q with probability q and 0 otherwise, with Y either exponentially distributed or Weibull distributed. Moreover, if Y is a NWU distribution, so is X. Throughout this paper we fix X to have unit mean, which in the case of degenerate distributions implies that also Y has unit mean. For the Weibull distribution, we simply fix $\kappa = 1/\Gamma(1/\alpha + 1)$ so that the mean equals 1 for any value of α.

3 Stochastic comparison results

In this section, we analyze how the scheduling policy (Section 3.1) and the copy correlation structure (Section 3.2) affect the performance of the system.

3.1 Comparison of scheduling policies

For a given system, we compare the total number of jobs with respect to the scheduling policy implemented in the system. We consider i.i.d. copies with exponential, NWU, and NBU service times, and then investigate scheduling policies with identical copies.

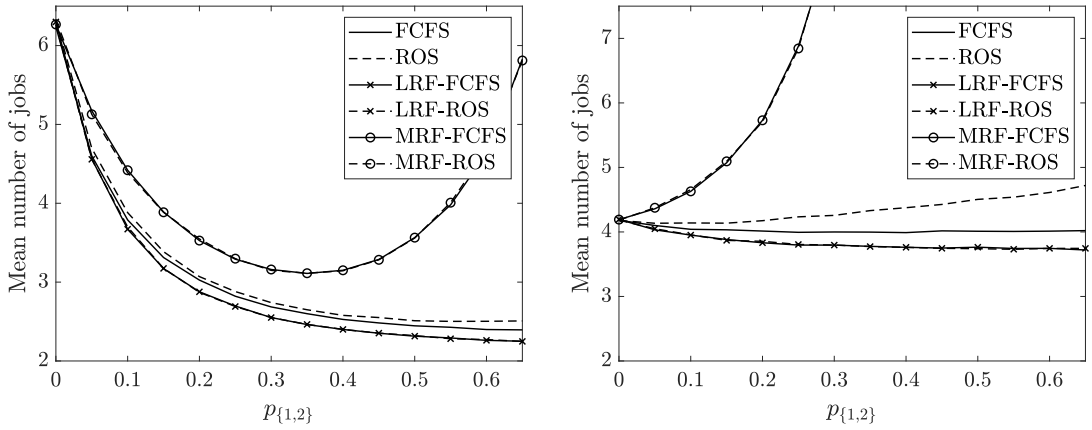
3.1.1 I.i.d. copies and exponential service times

We first assume that service times are exponentially distributed with i.i.d. copies. Due to the memoryless property, we can show that the number of jobs is insensitive to the implemented second-level policy Π_2.

Lemma 1. *Consider a redundancy system with a general topology and heterogeneous server capacities, where jobs have exponentially distributed service times and i.i.d. copies. Then, for any* Π_2 *and* Π_2', $\{N^{\Pi_1-\Pi_2}(t)\}_{t\geq 0} =_{st} \{N^{\Pi_1-\Pi_2'}(t)\}_{t\geq 0}$, *where* Π_1 *is a strict preemptive priority policy.*

Proof: Since Π_1 is a strict priority policy, exactly one job class has priority at any given time in a given server. Because of the i.i.d. copies assumption and exponentially distributed service times, within a job class, all non-idling policies are equivalent. □

This result holds because the first-level policy Π_1 is a strict, redundancy-aware, priority policy. In Figure 2, we show that if this first-level policy Π_1 is redundancy-oblivious, the redundancy-oblivious scheduling discipline does have an impact on the number of jobs in the system. We consider a W-model with $p_{\{1\}} = 0.35$ and $p_{\{2\}} = 1 - p_{\{1\}} - p_{\{1,2\}}$ and vary $p_{\{1,2\}}$. We observe in the figure that the mean number of jobs under Π_1-FCFS and Π_1-ROS coincide for both Π_1=LRF and, Π_1=MRF. However, the redundancy-oblivious policies FCFS and ROS provide different mean



(a) $\vec{\mu} = (1, 1)$ and $\lambda = 1.3$

(b) $\vec{\mu} = (1, 2)$ and $\lambda = 2$

Figure 2: The mean number of jobs for the W-model, with $p_{\{1\}} = 0.35$ and $p_{\{2\}} = 1 - p_{\{1\}} - p_{\{1,2\}}$, exponential service times and i.i.d. copies.

numbers of jobs in the system. These policies treat all classes equally, and hence Π_1 is redundancy-oblivious, so that Lemma 1 does not apply. We observe that FCFS outperforms ROS for any value of $p_{\{1,2\}}$. We further note that the differences between FCFS and ROS are most pronounced when the servers are heterogeneous (Figure 2 (b)) and when $p_{\{1,2\}}$ approaches $1 - p_{\{1\}}$.

We also observe in Figure 2 that LRF-Π_2, with Π_2=FCFS, ROS, outperform the other policies. This is consistent with the proposition below, which generalizes the result in [11] for LRF-FCFS.

Proposition 2. *Consider a redundancy system with a nested topology and heterogeneous server capacities where jobs have exponentially distributed i.i.d. copies. Then,*

$$\{N^{LRF-\Pi_2}(t)\}_{t\geq 0} \leq_{st} \{N^{\pi}(t)\}_{t\geq 0},$$

for any Π_2 *and any* π.

Proof: In [11] it was proven that $\{N^{LRF-FCFS}(t)\}_{t\geq 0} \leq_{st} \{N^{\pi}(t)\}_{t\geq 0}$ for any policy π. Together with Lemma 1 this gives the result. □

We note that for non-nested topologies, an optimal policy is expected to be more complex because an optimal choice of which class to serve will depend on the number of jobs in each class. Indeed, in Figure 3 we simulated a $K = 4$ server system with homogeneous capacities $\vec{\mu} = (1, 1, 1, 1)$ and a non-nested redundancy topology $\mathcal{C} = \{\{1, 2\}, \{1, 3\}, \{4\}, \{3, 4\}, \{2, 3, 4\}\}$, with $\vec{p} = (0.15, 0.15, 0.3, 0.15, 0.25)$. We observe that FCFS outperforms LRF-FCFS when the load in the system is sufficiently large.

3.1.2 I.i.d. copies and NWU service times

When jobs have i.i.d. copies and NWU service time distributions, the service time of a copy that is already in service is stochastically larger than that of an i.i.d. copy that has not received service yet. Hence, this suggests that whenever a server becomes available to a class, it will be better to serve a copy of a job that has already a copy elsewhere in service, to have it leave faster. That is exactly what policy Π_2=FCFS does. In the result below we show that, given a first-level policy Π_1, FCFS at the second level is indeed optimal. We note that in [20] this result was proved for the redundancy system with only one class of jobs. The proof is deferred to Appendix A.

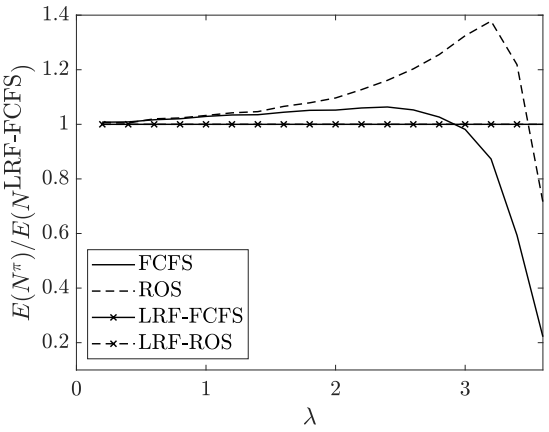


Figure 3: The relative mean number of jobs for a non-nested redundancy model with 4 servers, exponential service times and i.i.d. copies.

Proposition 3. *Consider a redundancy system with a general topology, heterogeneous server capacities, NWU service times and i.i.d. copies. Then,*

$$\{N^{\Pi_1-FCFS}(t)\}_{t\geq 0} \leq_{st} \{N^{\Pi_1-\Pi_2}(t)\}_{t\geq 0},$$

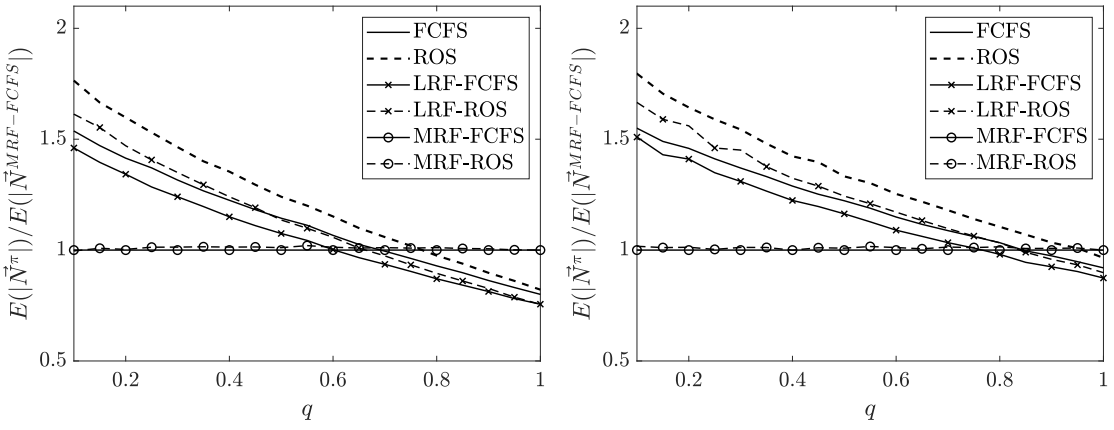
for all $t \geq 0$ *and any second-level policy* Π_2.

As an illustration, in Figure 4 we simulate the W-model with $\lambda = 1.5$, homogeneous capacities $\vec{\mu} = (1, 1)$ and $p_c = 1/3$ for all $c \in \mathcal{C}$. We assume that the service time distribution X is a mixture of Y/q with probability q and 0 otherwise, where $Y \sim F$ is NWU. In Figure 4 (a) we chose Y to be exponential. In Figure 4 (b) and (c) we chose Y to be Weibull. The squared coefficient of variation of X equals $C^2 = \frac{\mathbb{E}(Y^2)}{q\mathbb{E}(Y)^2} - 1$ and increases without bound when $q \to 0$. We note that in the special case where Y is exponentially distributed, we have $C^2 = 2/q - 1$. Consistent with Proposition 3, we observe that Π_1-FCFS (solid line) outperforms Π_1-ROS (dashed line) for both Π_1=LRF ($\times$) and Π_2=MRF ($\circ$). This observation also holds for single-level redundancy-oblivious policies, i.e., FCFS outperforms ROS, however, we did not obtain a proof for this.

In Figure 4 (a) and (b) we also observe that as q approaches 1, LRF-FCFS outperforms the other policies. In fact, when $q = 1$ and Y is exponentially distributed, it was shown in Proposition 2 that LRF-FCFS minimizes the number of jobs. When q approaches 0, that is $C^2 \to \infty$, we observe that MRF-FCFS outperforms all other scheduling policies. This example shows that there is not a unique first-level policy that minimizes the total number of jobs in the system for non-exponential service times.

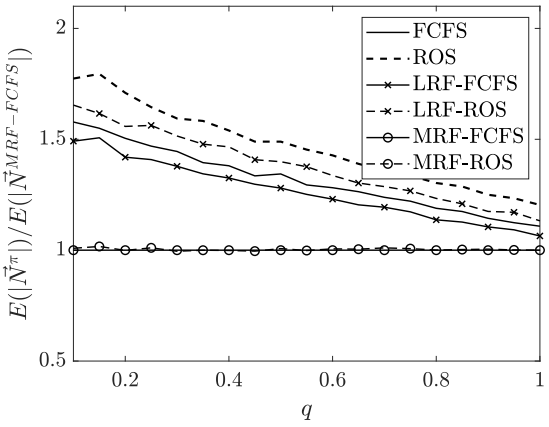
Under NWU service times, MRF might perform well because this policy serves at all time the copies of the job that has the most copies. However, given we consider nested systems, LRF is the first-level policy that minimizes the time that a server is idle. Hence, there is a trade-off, and which policy is optimal will strongly depend on the coefficient of variation of the service time distribution, which impacts how beneficial it is to serve copies of the same job (the more variable services are, the more profitable to serve multiple copies). The proposition below supports our observation for a K server system where each server has dedicated traffic and there is one flexible class of jobs that sends copies to all servers. The proof can be found in Appendix A.

Proposition 4. *Consider* K *heterogeneous servers with capacities* μ_s *where each server has a dedicated job class and there is an additional job class that sends copies to all the servers. That*



(a) $Y \sim$ exponential

(b) $Y \sim$ Weibull: $\alpha = 0.75$, $C_Y^2 = 1.83$



(c) $Y \sim$ Weibull: $\alpha = 0.5$, $C_Y^2 = 5$.

Figure 4: The mean number of jobs for the W-model with i.i.d. copies when $\vec{\mu} = (1, 1)$, $\lambda = 1.5$ and $p_c = 1/3$ for all $c \in \mathcal{C}$ with X a mixture of Y (top left) exponential, (top right) Weibull with parameters $\alpha = 0.75$ and $\kappa = 0.83$, and (bottom) Weibull with parameters $\alpha = 0.5$ and $\kappa = 0.5$.

is, $\mathcal{C} = \{\{s\}_{s\in S}, S\}$. We assume that the service time distribution X is a mixture of Y/q with probability q and 0 otherwise, where Y is NWU. Then,

$$q\mathbb{E}(N^{\mathrm{MRF-FCFS}}) < q\mathbb{E}(N^{\mathrm{LRF-FCFS}}) + o(1), \text{ as } q \to 0.$$

Combining Proposition 4 with Proposition 3, implies that for the heterogeneous server system with $\mathcal{C} = \{\{s\}_{s\in S}, S\}$, i.i.d. copies, and X being a mixture of Y/q with probability q and 0 otherwise, where Y is NWU, the MRF-FCFS policy is better than any other two-level policy as $q \to 0$.

3.1.3 I.i.d. copies and NBU service times

In the case of NBU service times and i.i.d. copies, we are only able to obtain partial results. This is not surprising, given that already when all jobs are of the same class, the only known results are

for the case of two homogeneous servers [20], or for an arbitrary number of servers but a saturated system [19]. We will focus on the two-server nested system, i.e., a W-model.

The following proposition gives a partial characterization for an optimal second-level policy. The proof can be found in Appendix A.

Proposition 5. *Consider a W-model with heterogeneous servers, NBU service times and i.i.d. copies. The first-level policy is hence either LRF or MRF. Whenever the first-level priority policy gives priority to class $\{1,2\}$ and the second-level policy decides not to idle, it is stochastically optimal to serve a copy of a class-$\{1,2\}$ job that has no copy yet in service in the other server (if possible). Server s should never idle when class $\{s\}$ has priority and there are jobs of class $\{s\}$ present, for $s = 1, 2$.*

The previous proposition does not completely characterize the optimal second-level policy. In particular, it does not tell us what should be done at a server s if there is only one class-$\{1,2\}$ job that has already received some service in the other server.

We further note that the proof of the previous proposition can not be generalized to more than two servers. The reason for this is given at the end of the proof in Appendix A.

The proposition above assumes that we know whether a class-$\{1,2\}$ job has received service at the other server. In practice, this information might not be available. In the following proposition, we therefore provide a comparison between FCFS and ROS and show that given the decision to serve a class-$\{1,2\}$ job, it is better to choose according to ROS, as this chooses with a higher chance a copy that has not received service elsewhere yet. The proof can be found in Appendix A.

Corollary 6. *Consider a W-model with heterogeneous servers, NBU service times and i.i.d. copies. The first-level policy is hence either LRF or MRF. Whenever the first-level priority policy serves class $\{1,2\}$ without idling, it is stochastically better for the second-level policy to serve according to ROS than FCFS.*

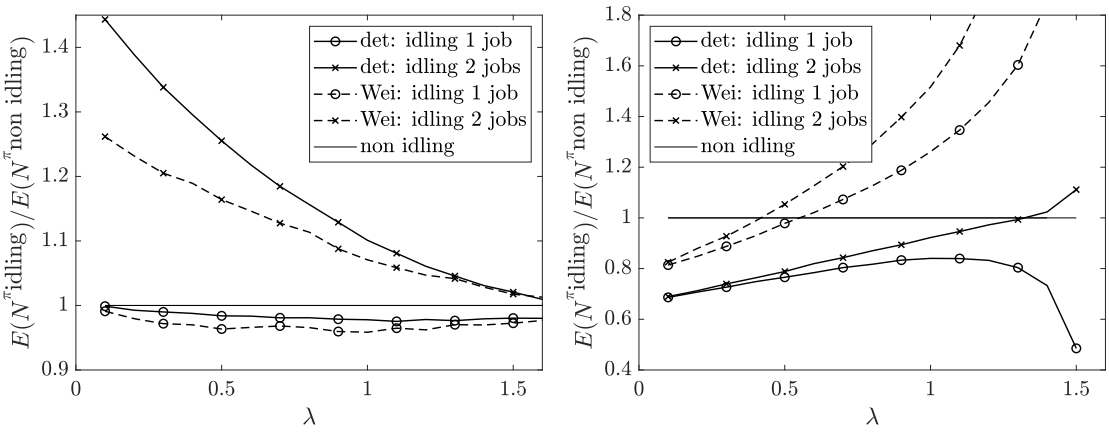
Note that the above proposition does not say whether or when non-idling or idling is optimal. Idling may be optimal because we do not allow preemption at the second level. We expect the optimal idling policy under Π_1-ROS to depend on the number of class-$\{1,2\}$ jobs since it might be profitable to idle a server when there is only one class-$\{1,2\}$ job which is already in service on the other server, in anticipation of a new class-$\{1,2\}$ job for this server that might be scheduled on this server. In Figure 5 we consider the W-model and simulate the performance of LRF-ROS (a) and MRF-ROS (b) when server 1 idles when according to the first-level policy it should start serving a new class-$\{1,2\}$ job, but there are only x class-$\{1,2\}$ jobs present, with x equal to either 1 or 2. For LRF-ROS (a) we observe that idling when there is only 1 class-$\{1,2\}$ job present is better than non-idling. For MRF-ROS we observe that for small enough arrival rate λ, the policies that idle when there are 1 class-$\{1,2\}$ jobs or up to 2 class-$\{1,2\}$ jobs are better than non-idling. However, as λ increases the performance of these policies strongly depends on the service time distribution.

In Figure 6 we compare the different policies (without idling) and observe that for a given first-level policy, ROS outperforms FCFS.

We do not have a comparison result for the first-level policies. Numerically we did observe though that LRF outperforms MRF for both deterministic and Weibull service times, see Figure 6. In the case of deterministic service times, we can indeed prove this. It follows directly from Proposition 7 with deterministic services.

3.1.4 Identical copies

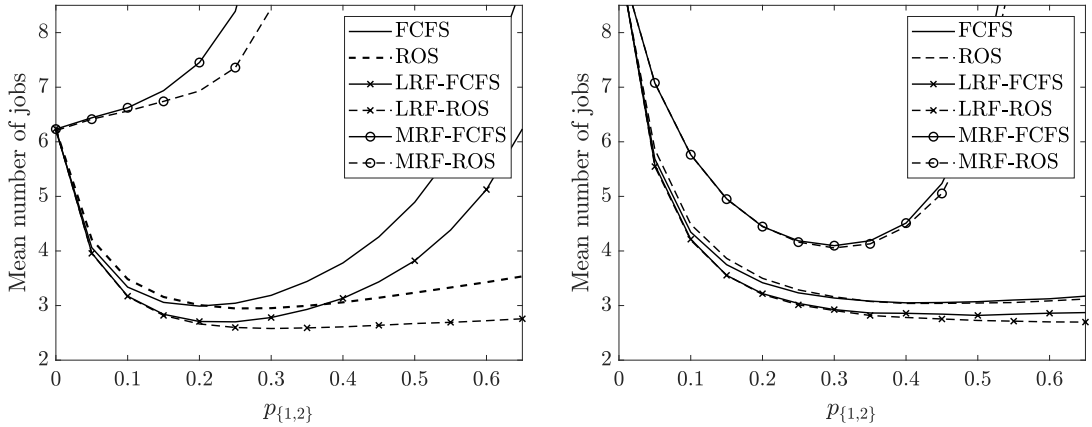
In this section we consider identical copies and general service times and investigate the impact of the first-level and second-level policies.



(a) LRF-ROS

(b) MRF-ROS

Figure 5: The mean number of jobs for the W-model with i.i.d. copies and with respect to λ, with capacities $\vec{\mu} = (1, 1)$, $p_c = 1/3$, and NBU service times, either deterministic (det, full line) or Weibull with $\alpha = 1.25$ (Wei, dashed line).



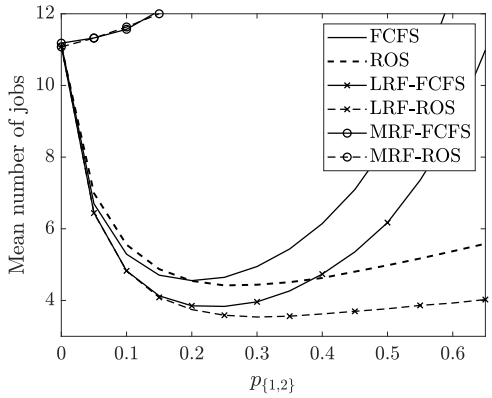
(a) Deterministic

(b) Weibull with $\alpha = 1.25$

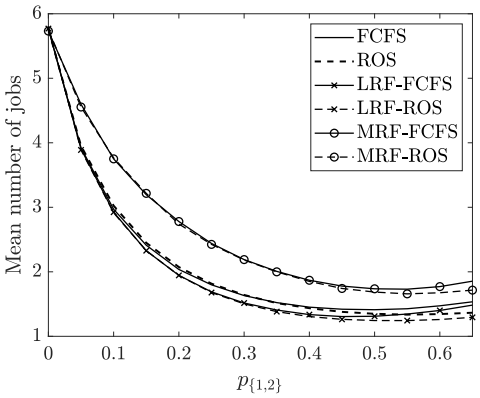
Figure 6: The mean number of jobs for the W-model with i.i.d. copies and capacities $\vec{\mu} = (1, 1)$, $\lambda = 1.4$, $p_{\{1\}} = 0.35$ and $p_{\{2\}} = 1 - p_{\{1\}} - p_{\{1,2\}}$.

In Figure 7, we plot the mean number of jobs under different policies for the W-model with identical copies and several choices of the service time distributions. We observe that LRF outperforms MRF for a given service time distribution and second-level policy. This can be explained as follows. When copies are identical, having several copies of the same job in service implies that capacity of one of the servers is unnecessarily dedicated to this job. Since LRF minimizes the number of copies of the same job in service, one would expect LRF to be optimal. In the result below, we show that this can be proved under certain conditions when the second-level policy is FCFS. The proof of the result can be found in Appendix A and follows by upper bounding the LRF system and using stochastic coupling arguments.

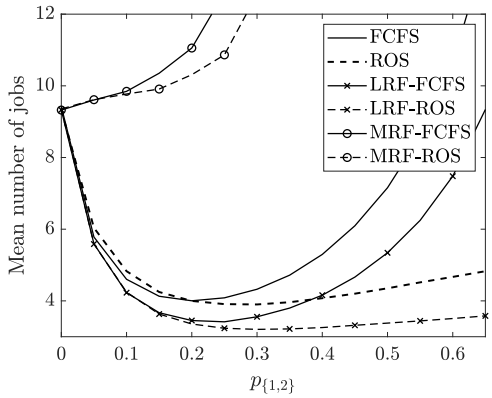
Proposition 7. *Consider K heterogeneous servers with capacities μ_s where $\mu_1 = \max_{s\in S}\{\mu_s\}$*



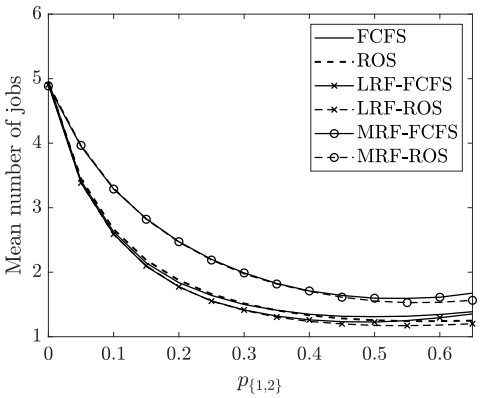
(a) Exponential, $\lambda = 1.4, \vec{\mu} = (1, 1)$



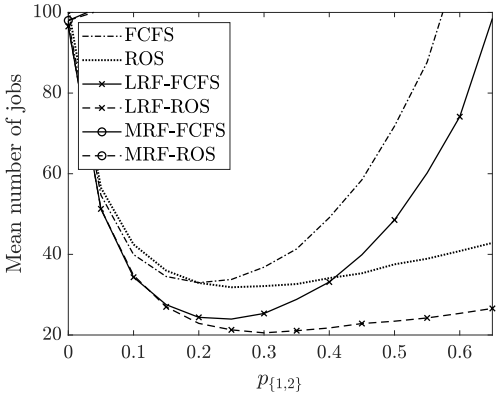
(b) Exponential, $\lambda = 1.3, \vec{\mu} = (2, 1)$



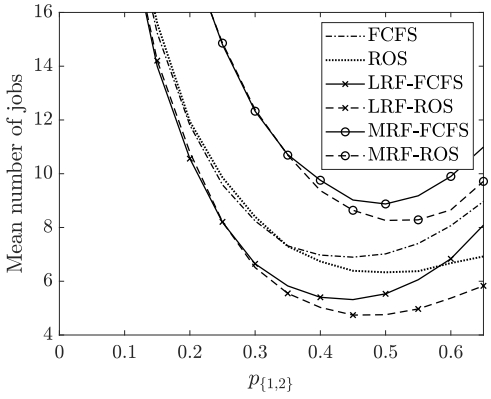
(c) Weibull, $\alpha = 1.25$ (NBU), $\lambda = 1.4, \vec{\mu} = (1, 1)$



(d) Weibull, $\alpha = 1.25$ (NBU), $\lambda = 1.3, \vec{\mu} = (2, 1)$



(e) Mixture exp., $q = 0.1$ (NWU), $\lambda = 1.4, \vec{\mu} = (1, 1)$



(f) Mixture exp., $q = 0.1$ (NWU), $\lambda = 1.3, \vec{\mu} = (2, 1)$

Figure 7: The mean number of jobs for the W-model with $p_{\{1\}} = 0.35$, $p_{\{2\}} = 1 - p_{\{1\}} - p_{\{1,2\}}$ and identical copies.

and each server has a dedicated job class and there is an additional job class that sends copies to

all the servers. That is, $\mathcal{C} = \{\{s\}_{s\in S}, S\}$. We assume general service times and identical copies. Assume that $p_S \geq p_{\{1\}}$ or that $\lambda \leq \lambda_0$, where λ_0 is given by Equation (11) *in Appendix A. Then it holds that*

$$\mathbb{E}(N^{LRF-FCFS}) \leq \mathbb{E}(N^{MRF-FCFS}).$$

For the second-level policy, we observe in Figure 7 that ROS performs better than FCFS. In the case of MRF as the first-level policy, we have a more general result, that is, MRF-FCFS is worse than any other MRF-Π_2 policy. The proof is deferred to the Appendix A.

Proposition 8. *Consider a redundancy system with a nested topology and heterogeneous server capacities, where the service times of the jobs are distributed according to a general distribution and copies are identical. Then, $\{M^{MRF-FCFS}(t)\}_{t\geq 0} \geq_{st} \{M^{MRF-\Pi_2}(t)\}_{t\geq 0}$ with preemptive MRF and where Π_2 is non-idling.*

The key property to prove the above result is that in a nested redundancy system with preemptive MRF-FCFS, all copies of a job enter service simultaneously and complete service in the server with highest capacity, wasting resources on the other servers serving this job. Hence, the system behaves as if each job is always served by its compatible server with the highest capacity, while the other compatible servers effectively idle until that job has departed.

The above result gives the worst second-level policy. It is however less clear what is the best second-level policy. We obtained the following partial characterizations of the optimal second-level policy, restricted to the W-model only. These partial characterizations coincide with those of Proposition 5 and Corollary 6 for i.i.d. copies and NBU service times, and their proof is the same.

Proposition 9. *Consider a W-model (so the first-level priority policy is either LRF or MRF) with heterogeneous servers and identical copies. Whenever the first-level priority policy gives priority to class-$\{1,2\}$ and the second-level policy decides not to idle, it is stochastically optimal to choose to serve a copy of a class-$\{1,2\}$ job that has no copy yet in service in the other server (if possible). Server s should never idle when class $\{s\}$ has priority and there are jobs of class $\{s\}$ present, for $s = 1, 2$.*

Corollary 10. *Consider a W-model with heterogeneous servers, general service times and identical copies. Whenever the first-level priority policy serves class $\{1,2\}$ without idling, it is stochastically better for the second-level policy to serve according to ROS than FCFS.*

We note that for deterministic service times, there is no distinction between i.i.d. and identical copies. Hence, combining Figures 6 and 7, we observe that for a given policy π, the performance deteriorates as the variability of the service time increases.

3.2 Comparison between i.i.d. copies and identical copies

In the present section we investigate how the correlation structure among the copies affects the performance of the system. For this section we define $N(t)$ as the total number of jobs in the system with i.i.d. copies, and $M(t)$ as the total number of jobs with identical copies. The proofs in this section are deferred to Appendix A.

In [7] the authors prove that the stability condition for the redundancy-d (with $d > 1$) model is larger under i.i.d. copies than under identical copies for FCFS. This is due to the fact that, when service times are exponential and copies are i.i.d., the departure rate of a subset of busy servers is given by the sum of all the service rates, whereas under identical copies it is given by the sum of the rates of servers giving service to different jobs. Hence, the departure rate under i.i.d. copies is at least as large as that under identical copies. In the following lemma, for a given policy Π_1-FCFS, we show a stronger result. For general service times and a general redundancy system, we show that the jobs with i.i.d. copies leave no later than with identical copies.

Proposition 11. *Consider a redundancy system with a general topology and heterogeneous server capacities, where the service times of the jobs are sampled from a general distribution. Then, for any preemptive policy* Π_1*, we have that* $\{N^{\Pi_1-FCFS}(t)\}_{t\geq 0} \leq_{st} \{M^{\Pi_1-FCFS}(t)\}_{t\geq 0}$.

We note that the previous proposition also holds when Π_1 is *not* a *strict* priority policy. In particular, setting Π_1 such that all job classes have the same priority, we obtain that $\{N^{\text{FCFS}}(t)\}_{t\geq 0} \leq_{st} \{M^{\text{FCFS}}(t)\}_{t\geq 0}$.

We believe that the above result would hold as well for other policies than Π_1-FCFS. We do however not have a proof for this. We note that when comparing Figure 6 (b) and Figure 7 (c) (both for the W-model with a Weibull distribution), the mean number of jobs under i.i.d. copies is smaller than under identical copies.

4 Stability condition

In the present section we discuss the stability condition for our scheduling policies. In the particular cases where either FCFS or ROS is implemented and in the absence of a first-level policy, the stability condition has been characterized under various conditions, which we briefly summarize in Section 4.1. These stability results are however no longer valid when a first-level policy is implemented. In Section 4.2 we discuss stability results for first-level policies LRF and MRF. We further note that Section 4 is focussed on exponential service times. To the best of our knowledge, no explicit stability results have been obtained so far for ROS or FCFS when assuming general service times.

4.1 Stability conditions for single-level FCFS and ROS

Under the FCFS service policy, the stability condition has been fully characterized in the case where jobs have exponentially distributed i.i.d. copies, [16].

Proposition 12. *[16] The redundancy heterogeneous server system with exponential job sizes and i.i.d. copies under FCFS, is stable if, for all* $C \subseteq \mathcal{C}$*,*

$$\lambda \sum_{c\in C} p_c < \sum_{s\in S(C)} \mu_s, \tag{1}$$

where $S(C) = \cup_{c\in C}\{s \in c\}$.

This stability condition coincides with that of the cancel-on-start redundancy system with exponential service times and FCFS. In addition, for exponential service times, Equation (1) is the maximum stability condition, i.e., there does not exist a policy (with or without redundancy) that makes the system stable if one of the inequalities of (1) were not satisfied.

When copies are identical, the stability region under redundancy is reduced under FCFS. For example, for the redundancy-d model with homogeneous servers, the maximum stability condition (1) simplifies to $\lambda < \mu K$ for i.i.d. copies, but for identical copies we have the stricter condition below [5, 7].

Proposition 13. *[7] The redundancy-*d *model under FCFS with exponential job sizes and identical copies is stable if* $\lambda < \bar{\ell}\mu$ *and unstable if* $\lambda > \bar{\ell}\mu$*, where* $\bar{\ell}$ *denotes the mean number of jobs in service in the associated saturated system.*

Under ROS, in [7] the authors prove that for the redundancy-d model under either i.i.d. copies or identical copies, the stability region is not reduced due to adding redundant copies.

Theorem 14. *The redundancy-*d *model where ROS is implemented, jobs have exponentially distributed service times and either i.i.d. copies or identical copies, is stable if* $\lambda < K\mu$*.*

4.2 Stability condition under LRF and MRF

I.i.d. copies

We first assume the nested redundancy topology and that copies are i.i.d.. The stability condition under preemptive LRF-Π_2 with exponentially distributed service times is straightforward from Proposition 2 and [16]. In particular, the result shows that preemptive LRF-Π_2 is maximally stable. Below, we prove that the same holds true for non-preemptive LRF-Π_2.

Proposition 15. *Consider a redundancy system with nested topology, where jobs are exponentially distributed with unit mean and have i.i.d. copies, and LRF-Π_2 is implemented. When the LRF policy is either preemptive or non-preemptive, the system is stable if for all $c \in \mathcal{C}$,*

$$\lambda \sum_{\tilde{c} \subseteq c} p_{\tilde{c}} < \sum_{s \in c} \mu_s. \tag{2}$$

The system is unstable if there exists $\hat{c} \in \mathcal{C}$ such that $\lambda \sum_{\tilde{c} \subseteq \hat{c}} p_{\tilde{c}} > \sum_{s \in \hat{c}} \mu_s$.

We note that for a nested topology, the maximum stability condition as in (1) simplifies to (2). *Proof:* The stability result when the LRF policy is preemptive follows directly from Proposition 2 and [16]. From Proposition 2, the stability region under FCFS must be at least as large as that under FCFS. For exponential service times, the latter is maximally stable [16], and hence, so is LRF-Π_2 with preemptive LRF. The proof when LRF policy is non-preemptive can be found in Appendix B. □

The assumption that the redundancy topology is nested is crucial for maximum stability to hold. To see this we refer to the following example where we consider a non-nested redundancy model and observe that LRF-FCFS and LRF-ROS are not maximally stable.

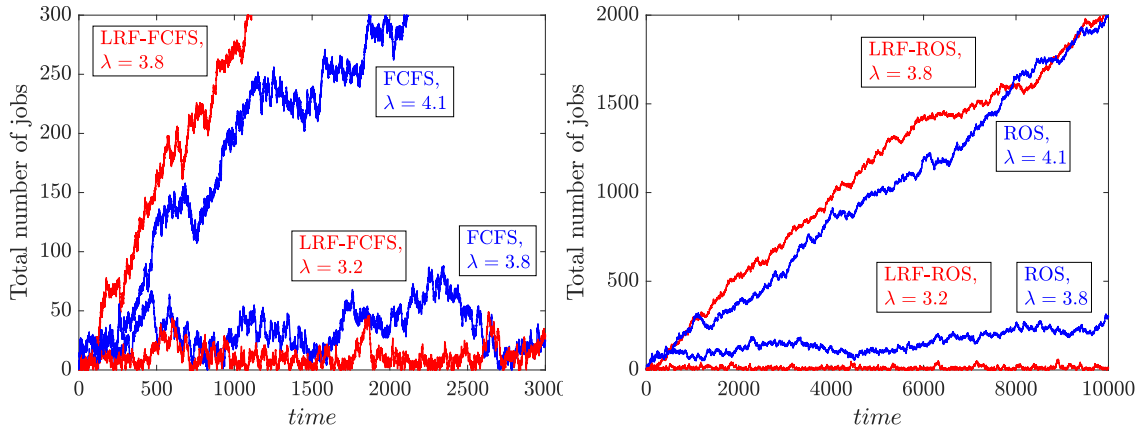
Example 1. **Non-nested model:** Consider a $K = 4$ server system with homogeneous capacities $\mu_s = 1$, $s = 1, \ldots, 4$, and with the following redundancy topology: $\mathcal{C} = \{\{1,2\}, \{1,3\}, \{4\}, \{3,4\}, \{2,3,4\}\}$. We note that this particular topology is non-nested. We chose $\vec{p}$ such that the maximum achievable stability region is $\lambda < 4$. In Figure 8, we plot the trajectory of the total number of jobs as a function of time for various loads and service policies. In the figure we observe that neither LRF-FCFS nor LRF-ROS are stable when $\lambda = 3.8$, while FCFS and ROS do provide a stable system for $\lambda = 3.8$. We include plots with $\lambda = 3.2$ and $\lambda = 4.1$ for comparison.

The assumption that Π_1=LRF is crucial in order for the maximum stability result to hold. To see this, we refer to Example 2 where we will show that the nested N-model is not maximally stable under either MRF-FCFS or MRF-ROS.

Example 2. **N-model:** We assume the N-model where servers have heterogeneous capacities $\vec{\mu} = (\mu_1, \mu_2)$. The maximum stability condition as given in (1) when jobs have exponentially distributed service times simplifies to $\lambda p_{\{2\}} < \mu_2$ and $\lambda < \mu_1 + \mu_2$.

In the case of MRF-Π_2, with Π_2 non-idling, we have that class-$\{2\}$ jobs can only be served if there is no class-$\{1,2\}$ job present in the system. Let us denote by $\mu_{\{1,2\}}$ the mean departure rate of class-$\{1,2\}$ jobs in the system and by $\rho_{\{1,2\}} = \lambda(1 - p_{\{2\}})/\mu_{\{1,2\}}$. Because copies are i.i.d., $\mu_{\{1,2\}} = \mu_1 + \mu_2$, so the stability condition of class $\{1,2\}$ is $\lambda p_{\{1,2\}} < \mu_1 + \mu_2$. Now class $\{2\}$ can only be served a $(1 - \rho_{1,2})$ fraction of the time. Thus, the stability condition for class $\{2\}$ is given by $\lambda p_{\{2\}} < \mu_2(1 - \rho_{1,2})$. This is a more strict condition than the maximum stability condition that only required $\lambda p_{\{2\}} < \mu_2$.

In general, in order for a policy to be stable under the maximum stability condition, the policy needs to correctly balance the jobs over the different heterogeneous servers so that in the long run, each class is in service an appropriate fraction of time. In general, this can be difficult. However,



(a) FCFS and LRF-FCFS

(b) ROS and LRF-ROS

Figure 8: A non-nested redundancy topology with homogeneous server capacities. The trajectory of the total number of jobs for exponentially distributed service times and i.i.d. copies for scheduling policies Π_1-Π_2 with Π_1=LRF, and Π_2=FCFS, ROS for various arrival rates.

when the topology is nested, a job class only shares servers with classes that are subsumed within. As a result, when implementing LRF, each job class receives full capacity on its feasible servers when no higher priority jobs are present.

Copies with general correlation structure

In the following proposition we discuss the stability condition when copies follow a general correlation structure. We first show for a nested topology that LRF-ROS is maximally stable. The proof is deferred to Appendix B.

Proposition 16. *Consider a redundancy system with nested topology, where jobs are exponentially distributed with unit mean and copies follow some general correlation structure. Under LRF-ROS, with LRF either preemptive or non-preemptive, the system is stable if for all $c \in \mathcal{C}$,*

$$\lambda \sum_{\tilde{c} \subseteq c} p_{\tilde{c}} < \sum_{s \in c} \mu_s.$$

The system is unstable if there exists $\hat{c} \in \mathcal{C}$ such that $\lambda \sum_{\tilde{c} \subseteq \hat{c}} p_{\tilde{c}} > \sum_{s \in \hat{c}} \mu_s$.

For a nested redundancy topology with LRF, given the state of the system, the job class in service at each server is completely characterized. Furthermore, since the second-level policy is ROS, when the number of jobs in service is large, the probability that more than one copy of the same job is simultaneously in service is close to zero, regardless of the correlation structure among the copies. Hence, we can completely characterize the instantaneous departure rate of the system when in the fluid limit. We note that for the redundancy-oblivious policy ROS, the stability condition is unknown so far (with the exception of the redundancy-d model).

For non-nested systems, or two-level policies other than LRF-ROS, we did not succeed in deriving the stability conditions. We do however show, in the examples below, some situations that might not be maximally stable.

In the following example we discuss the stability condition for a nested model, but under two-level policies other than LRF-ROS. We observe that these systems are not maximally stable.

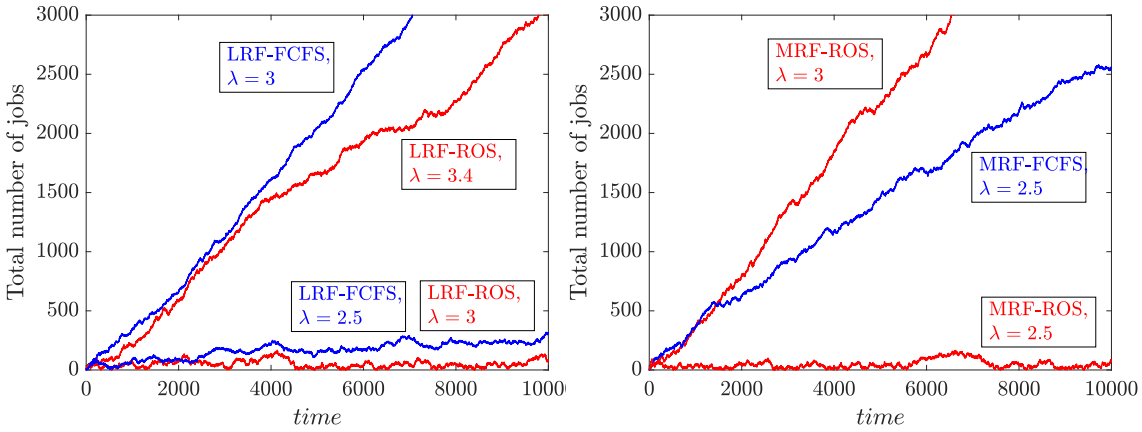


Figure 9: Trajectory of the redundancy system of Example 4 under different two-level policies and arrival rates.

Example 3. N**-model:** We consider the N-model where servers have heterogeneous capacities $\vec{\mu} = (\mu_1, \mu_2)$ as in Example 2. We recall that the maximum stability condition when jobs have exponentially distributed service times is given by $\lambda p_{\{2\}} < \mu_2$ and $\lambda < \mu_1 + \mu_2$.

In the case of **MRF-**Π_2 with identical copies and Π_2 non-idling, the departure rate of class-$\{1,2\}$ is given by $\mu_{\{1,2\}} = \max\{\mu_1, \mu_2\}$. Because class $\{2\}$ can only be served a $(1 - \rho_{\{1,2\}})$ fraction of the time, the stability condition is given by $\lambda p_{\{2\}} < \mu_{\{1,2\}}(1 - \rho_{\{1,2\}})$, which is more strict than the maximum stability condition.

In the case of **LRF-FCFS** with identical copies, when class $\{2\}$ and class $\{1,2\}$ are both present in the system, the total departure rate is given by $\mu_1 + \mu_2$. Then, when class $\{2\}$ is not present (which happens $\rho_2 = \lambda p_{\{2\}}/\mu_2$ fraction of the time), the total departure rate is time-varying and given by $\mu_{\{1,2\}}(t) := \alpha(t)\mu_1 + (1 - \alpha(t))\mu_2$, where $\alpha(t)$ is either 0 or 1. Note that $\mu_{\{1,2\}}(t) \leq \max\{\mu_1, \mu_2\}$. Therefore, the stability condition is given by $\lambda p_{\{2\}}/\mu_2$, $\lambda < \mu_1 + \mu_2$ and $\lambda p_{\{1,2\}} < \tilde{\mu}_{\{1,2\}}(1 - \rho_2)$, where $\tilde{\mu}_{\{1,2\}} \leq \max\{\mu_1, \mu_2\}$, which is more strict than the maximum stability condition.

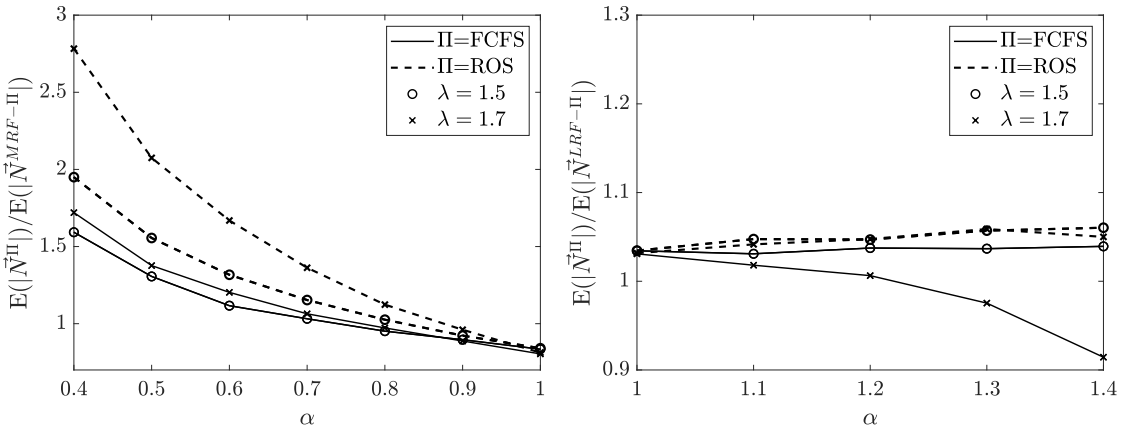
We did not succeed in obtaining the stability conditions for second-level policies other than LRF-ROS. We do however have the following comparison result, which is a direct consequence of Proposition 8.

Corollary 17. *Consider a redundancy system with nested topology, where jobs are exponentially distributed with unit mean and identical copies. The stability condition under preemptive MRF-ROS, is at least as large as that under preemptive MRF-FCFS.*

We note that the above corollary does not give us exact values for stability conditions, since for both MRF-ROS and MRF-FCFS, the stability condition is unknown.

We now consider a numerical example (Example 4) of a non-nested system under LRF-ROS with identical copies and observe that it is not maximally stable.

Example 4. **Non-nested model:** We consider 4 homogeneous servers with unit capacities, identical copies, and jobs dispatch either 2 or 3 copies chosen uniformly at random. The maximum stability condition is $\lambda < 4$, (1). However, in Figure 9 we observe that already for $\lambda = 3.4$ the system is not stable for any of the policies Π_1-Π_2, with $\Pi_1 \in \{LRF, MRF\}$ and $\Pi_2 \in \{FCFS, ROS\}$, including the LRF-ROS policy.



(a) $\alpha \leq 1$, NWU.

(b) $\alpha \geq 1$, NBU.

Figure 10: The mean number of jobs for the W-model with i.i.d. copies when $\vec{\mu} = (1, 1)$ and $\vec{p} = (1/6, 1/6, 2/3)$ with X the Weibull distribution with parameter α.

Regarding the stability condition, we can draw the following observations. We observe that the stability region under LRF-Π_2 is larger than that under MRF-Π_2. For LRF-ROS we observe that the stability region is at least $\lambda < 3$, whereas for MRF-ROS, we observe that already for $\lambda = 3$ the system is unstable. Similarly, we observe that the stability region for LRF-FCFS is at least $\lambda < 2.5$, whereas for MRF-FCFS already for $\lambda = 2.5$ the system is unstable. Second, in Figure 9 we observe that the stability region under Π_1-ROS is larger than that under Π_1-FCFS, which in the case of MRF was proved for nested topologies, see Corollary 17.

5 Redundancy-aware versus redundancy-oblivious scheduling

In the previous sections, we obtained several (partial) optimality results for two-level policies. In this section we will investigate whether these two-level policies are worth the hassle, that is, whether they can improve significantly the performance of the system compared to redundancy-oblivious schedulers. We focus in this section on nested topologies, as most optimality results are for that context, and it can model server pools in data centres.

5.1 I.i.d. copies

For NWU service times, we observe from Figure 4 for the W-model that the gap between Π_1-ROS and ROS, and the gap between Π_1-FCFS and FCFS, increases as the variability of the service time increases. That is, as q goes to zero, or as the distribution Y becomes more variable (i.e., moving from Figure 4(a) to 4(b) to 4(c)). The redundancy-oblivious policies can be more than a factor 1.5 worse than the MRF redundancy-aware version. To further investigate this, in Figure 10 (a) we plot the performance as a function of the Weibull parameter α, for $\alpha \leq 1$, that is for NWU distributions. As α decreases, the service time distribution becomes more variable. We compare the performance of redundancy-oblivious policies Π against redundancy-aware policies MRF-Π. We observe that redundancy-oblivious policies can have up to 2.5 times more jobs in the system than the MRF redundancy-aware policy for $\alpha = 0.4$.

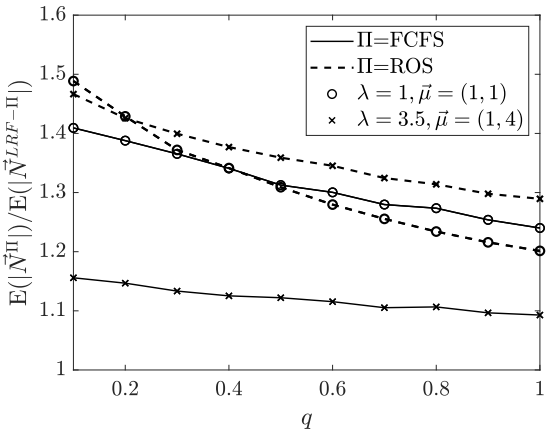


Figure 11: The mean number of jobs for the W-model with identical copies when $\vec{p} = (0.2, 0.2, 0.6)$ and with X mixture of exponential distribution (q) with respect to q.

On the other hand, for NBU service times, being redundancy aware matters less. For example, we can observe from Figure 6 that the gaps between ROS and LRF-ROS and between FCFS and LRF-FCFS are small, especially compared to the gap between ROS and FCFS. In Figure 10 (b) we plot the Weibull distribution for $\alpha \geq 1$ (hence NBU), and plot the ratio of the redundancy-oblivous policy Π against the redundancy-aware policy LRF-Π. Again, the redundancy-oblivious policy performs very similarly to the LRF redundancy-aware version.

5.2 Identical copies

For identical copies, we observed in Section 3.1.4 that LRF is an efficient first-level policy. We now first focus on Figure 7 and compare the performance of redundancy-aware policies LRF-Π_2 to that of redundancy-oblivious policies Π_2, for Π_2 =FCFS, ROS. We observe from Figure 7 that the mean number of jobs under LRF-Π_2 is smaller than that under Π_2 for any value of $p_{\{1,2\}}$. We also note that this gap is more pronounced when the variability of the service time distribution is large. Note that the exponential distribution has squared coefficient of variation $C^2 = 1$ (Figure 7 (a) and (b)), the Weibull distribution with $\alpha = 1.25$ has $C^2 = 0.64$ (Figure 7 (c) and (d)), and the degenerate hyperexponential distribution with $q = 0.1$ has $C^2 = 19$ (Figure 7 (e) and (f)). Redundancy-oblivious policies can be more than a factor 1.5 worse than their LRF redundancy-aware version.

In Figure 11 we plot the ratio between the mean number of jobs under a given policy Π and the mean number of jobs under LRF-Π, for Π = FCFS, ROS, when jobs have a mixture of exponential (q) distributed service times with respect to parameter q. We observe that the redundancy-oblivious policies are more than a factor 1.2 worse than the LRF redundancy-aware version. Moreover, we observe that this factor increases up to 1.5 as the variability of the service time distribution increases, that is, as q goes to 0.

6 Conclusions

We have explored, for nested systems and cancel-on-completion redundancy, the performance impact of two-level policies based on the level of redundancy, and compared these policies with traditional single-level service disciplines such as FCFS. Our theoretical and numerical results

indicate that FCFS is the best single-level policy and the best second-level policy when service times are NWU (so highly variable) and i.i.d. across copies. When variability is very high, it may be advantageous to use MRF as the first-level policy. When service time variability is low, or copies are identical, ROS is the best single-level policy and second-level policy, and LRF is the best first-level policy, though the first-level has less of an impact with low-variability services.

References

[1] Osman Akgun, Rhonda Righter, and Ronald Wolff. Multiple server system with flexible arrivals. *Applied Probability Trust*, 43: 985–1004, 2011.

[2] Osman Akgun, Rhonda Righter, and Ronald Wolff. Partial flexibility in routeing and scheduling. *Advances in Applied Probability*, 45: 673–691, 2013.

[3] Ganesh Ananthanarayanan, Ali Ghodsi, Scott Shenker, and Ion Stoica. Why let resources idle? aggressive cloning of jobs with dolly. In *Proceedings of the 4th USENIX Conference on Hot Topics in Cloud Computing*, HotCloud' 12, pp. 17, USA, 2012.

[4] Ganesh Ananthanarayanan, Ali Ghodsi, Scott Shenker, and Ion Stoica. Effective straggler mitigation: Attack of the clones. In *NSDI*, 13: 185–198, 2013.

[5] Elene Anton, Urtzi Ayesta, Matthieu Jonckheere, and Ina Maria Verloop. A survey of stability results for redundancy models. In *Alexey B. Piunovskiy and Yi Zhang (eds.), Modern Trends in Controlled Stochastic Processes: Theory and Applications, V.III.* Springer US, 2021.

[6] Elene Anton, Urtzi Ayesta, Matthieu Jonckheere, and Ina Maria Verloop. Improving the performance of heterogeneous data centers through redundancy.*Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS), SIGMETRICS 2021*, 4(3): Article 48, pp. 29, 2020.

[7] Elene Anton, Urtzi Ayesta, Matthieu Jonckheere, and Ina Maria Verloop. On the stability of redundancy models. *Operations Research* 69(5): 1540–1565, 2021.

[8] Thomas Bonald and Céline Comte. Balanced fair resource sharing in computer clusters. *Performance Evaluation*, 116: 70–83, 2017.

[9] Maury Bramson. *Stability of Queueing Networks*. Springer, 2008.

[10] Jeffrey Dean and Luiz André Barroso. The tail at scale. *Communications of the ACM*, 56(2): 74–80, 2013.

[11] Kristen Gardner, Mor Harchol-Balter, Esa Hyytia, and Rhonda Righter. Scheduling for efficiency and fairness in systems with redundancy. *Performance Evaluation*, 116: 1–25, 2017.

[12] Kristen Gardner, Mor Harchol-Balter, Alan Scheller-Wolf, and Benny van Houdt. A better model for job redundancy: Decoupling server slowdown and job size. *IEEE/ACM Transactions on Networking*, 25(6): 3353–3367, 2017.

[13] Kristen Gardner, Mor Harchol-Balter, Alan Scheller-Wolf, Mark Velednitsky, and Samuel Zbarsky. Redundancy-d: The power of d choices for redundancy. *Operations Research*, 65: 1078–1094, 2017.

[14] Kristen Gardner, Esa Hyytiä, and Rhonda Righter. A little redundancy goes a long way: Convexity in redundancy systems. *Performance Evaluation*, 131(4): 22–42, 2019.

[15] Kristen Gardner and Rhonda Righter. Product forms for fcfs queueing models with arbitrary server-job compatibilities: An overview, *Queueing Systems,* 96(1-2): 3–51, 2020.

[16] Kristen Gardner, Samuel Zbarsky, Sherwin Doroudi, Mor Harchol-Balter, Esa Hyytiä, and Alan Scheller-Wolf. Queueing with redundant requests: exact analysis. *Queueing Systems*, 83(3-4):227–259, 2016.

[17] Mor Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

[18] Gauri Joshi, Emina Soljanin, and Gregory Wornell. Queues with redundancy: Latency-cost analysis. *ACM SIGMETRICS Performance Evaluation Review*, 43(2): 54–56, 2015.

[19] Yusik Kim, Rhonda Righter, and Ronald Wolff. Job replication on multiserver systems. *Advances in Applied Probability*, 41: 546–575, 2009.

[20] Ger Koole and Rhonda Righter. Resource allocation in grid computing. *Journal of Scheduling*, 11: 163–173, 2008.

[21] Kangwook Lee, Nihar B. Shah, Longbo Huang, and Kannan Ramchandran. The mds queue: Analysing the latency performance of erasure codes. *IEEE Transactions on Information Theory*, 63(5):2822–2842, 2017.

[22] Leela Nageswaran and Alan Scheller-Wolf. Queues with redundancy: Is waiting in multiple lines fair?, *Manufacturing & Service Operations Management*, 2021.

[23] Youri Raaijmakers and Sem Borst. Achievable stability in redundancy systems, *Proc. ACM Meas. Anal. Comput. Syst.* 4(3): Article 46, 2020.

[24] Philippe Robert. *Stochastic Networks and Queues*. Springer-Verlag, 2003.

[25] Nihar B. Shah, Kangwook Lee, and Kannan Ramchandran. When do redundant requests reduce latency? *IEEE Transactions on Communications*, 64(2): 715–722, 2016.

[26] Ashish Vulimiri, Philip Brighten Godfrey, Radhika Mittal, Justine Sherry, Sylvia Ratnasamy, and Scott Shenker. Low latency via redundancy. In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, 283–294, 2013.

Appendix

A: Proofs of Section 3

Proof of Proposition 3: We prove that when there are two jobs of the same class to be served, it is always better to serve a copy of the job that already has a copy in service in another server, than to serve a copy of the job that has no other copies in service. That is a stronger result than the claim of the proposition. The argument is similar to that of [14], but we outline the proof here briefly for completeness.

We assume that at some time t, under policy π a server starts to serve a copy of a job that has no other copies in service, say this copy is from job 2 and in server 1. Additionally, there is a job in server 1 of the same class as job 2 that has copies being served in another server(s), say that this

is job 1. Let $\mathcal{A}$ be the set of servers that job 1 has already received some service on and let $\mathcal{B}$ be the set of servers that can serve job 1 but have not yet started to serve job 1.

We let π' serve job 1 on server 1 at time t and thereafter always serve job 1 whenever π serves job 2 or job 1 on any server in $\mathcal{B}$, until either job 1 or job 2 completes service under π, at some time τ. Otherwise we let π' agree with π until time τ (including when π serves job 1 on servers in $\mathcal{A}$). We couple the service times of the copies of jobs 1 and 2 on servers in $\mathcal{B}$ under the two policies, and let all other service times and arrival times be the same under both policies. At time τ, under policy π', job 1 has completed service and job 2 has not yet received service at any of its compatible servers. Under policy π, either job 1 or job 2 has completed service. Let us denote by a the job that did not complete service under π, a is either job 1 or job 2. We note that job a received service partially at some servers under policy π, say servers in the set $\mathcal{J}$. We couple the (new) service times of job 2 under π' on servers $\mathcal{J}$, denoted by X'_{2j} for $j = 1, \ldots, |\mathcal{J}|$, with the remaining service times of job a, denoted by X_{aj} for $j = 1, \ldots, |\mathcal{J}|$, so that, $X'_{2j} \leq X_{aj}$ with probability 1 for all $j = 1, \ldots, |\mathcal{J}|$. We can do this from the NWU assumption. We let π' serve a copy of job 2 whenever π serves a copy of job a from time τ on, until job 2 completes under π'. Thereafter, we let the servers that are serving a copy of job a in π idle in π', and otherwise let π' agree with π. Then $\{\vec{N}'(t)\}_{t\geq 0} \leq \{\vec{N}(t))\}_{t\geq 0}$ with probability 1.

Repeating the argument gives that FCFS that is possibly idling is optimal. Then, we are left to verify that the non-idling FCFS policy is better than with idling FCFS.

Assume that at time t, π idles a server for some time τ when it has copies to serve. Say this is server 1 that idles τ units of time and the next copy to serve in server 1 with highest priority is of job 1. We let π' agree with π for all of its copies correlation decisions, and let their arrival times and service times be coupled. Moreover, we let π' agree with π until time t. At time t, under π' server 1 serves the copy of job 1 from time t to time $t + \sigma$ where $\sigma = \min\{\tau, s, r - t, a - t\}$, s is the remaining service time of the copy of job 1 on server 1, r is the earliest completion time of all other copies of job 1 running on other servers and a is the arrival time of a higher priority job than job 1. We let π' agree with π for all other decisions between times t and $t + \sigma$. Three different events can occur at time σ:

- $\sigma = \tau$. We let π' idle server 1 after time σ whenever π serves job 1 on server 1, until it has received service for τ units of time in that server under π, at some time $\sigma' > \sigma$. Let π' otherwise agree with π for all time after σ and all servers. Then the systems under the two policies will be in the same states at time σ', and $\{N'(s)\}_{s\geq 0} = \{N(s)\}_{s\geq 0}$ wp 1.

- $\sigma = s$. We let π' idle the servers serving job 1 after time σ whenever π serves job 1 on those servers, until job 1 departs under π at time σ', say, and let π' otherwise agree with π. We let π' agree with π thereafter. At time σ', the two systems will be in the same state, but job 1 will have departed earlier under π', so $\{N'(s)\}_{s\geq 0} \leq \{N(s)\}_{s\geq 0}$ with probability 1.

- $\sigma = r - t$. Then the job departs at time σ in both systems and they will be in the same state. Letting π' agree with π after time σ, $\{N'(s)\}_{s\geq 0} = \{N(s)\}_{s\geq 0}$ with probability 1.

- $\sigma = a - t$. At time a, job 1 is preempted in π'. Then, we let π' idle server 1 after time a whenever π serves job 1 on server 1, until it has received service for $a - t$ units of time in that server (like job 1 received under policy π'). Let π' otherwise agree with π for all time after a. Then the systems under the two policies will be in the same states at time a, and $\{N'(s)\}_{s\geq 0} = \{N(s)\}_{s\geq 0}$ with probability 1.

□

Proof of Proposition 4: For ease of notation we remove the second-level policy from the superscript, which is FCFS under both systems. We note that for the random variable X, $\mathbb{E}(X) = \mathbb{E}(Y)$ and $\mathbb{E}(X^2) = \mathbb{E}(Y^2)/q$. Let us denote by $Z := \min_s\{Y_i/\mu_i\}$.

For a two-class single-server queue where class 1 has preemptive priority over class 2, the mean number of class-k jobs is given by [17]

$$\mathbb{E}(N(k)) = \lambda p_k \left[\frac{\mathbb{E}(X_k)}{1 - \sum_{i=1}^{k-1} \rho_i} + \frac{\sum_{i=1}^{k} \rho_i \frac{\mathbb{E}(X_i^2)}{2\mathbb{E}(X_i)}}{(1 - \sum_{i=1}^{k-1} \rho_i)(1 - \sum_{i=1}^{k} \rho_i)} \right], \; k = 1, 2, \tag{3}$$

where X_k is the service time distribution of class-k jobs, $\rho_k = \lambda_k \mathbb{E}(X_k)$, and $\lambda_k = p_k\lambda$ is the arrival rate of class-k jobs, $k = 1, 2$.

From the above result, we can compute the mean number of jobs under MRF-FCFS. We note that class-S jobs have preemptive priority over all other classes. Let us denote by $U = \min_s\{X_s/\mu_s\}$ the service time of class-S jobs under the MRF policy. Thus, $U = 0$ with probability $1 - q^K$ and $U = \frac{Z}{q}$ with probability q^K. The latter implies that $\mathbb{E}(U) = q^{K-1}\mathbb{E}(Z)$ and $\mathbb{E}(U^2) = q^{K-2}\mathbb{E}(Z^2)$. By applying Equation (3) we obtain

$$\begin{aligned} \mathbb{E}(N_S^{\mathrm{MRF}}) &= \lambda p_{\{S\}} \left(\mathbb{E}(U) + \frac{\lambda p_{\{S\}} \mathbb{E}(U^2)}{2(1 - \lambda p_{\{S\}} \mathbb{E}(U))} \right) \\ &= \lambda p_{\{S\}} \left(q^{K-1}\mathbb{E}(Z) + \frac{\lambda p_{\{S\}} q^{K-2} \mathbb{E}(Z^2)}{2(1 - \lambda p_{\{S\}} q^{K-1} \mathbb{E}(Z))} \right). \end{aligned} \tag{4}$$

Class-$\{s\}$ jobs are served if there are no class-S jobs present. Let us denote by W_s a generic service time of class-$\{s\}$ job, so $W_s = Y/\mu_s$, $\mathbb{E}(W_s) = \mathbb{E}(Y)/\mu_s$ and $\mathbb{E}(W^2) = \mathbb{E}(Y^2)/(q\mu_s^2)$. By applying Equation (3), we obtain

$$\begin{aligned} &\mathbb{E}(N_{\{s\}}^{MRF}) \\ &= \frac{\lambda p_{\{s\}}}{1 - \lambda p_S \mathbb{E}(U)} \left(\mathbb{E}(W_s) + \frac{\lambda p_S \mathbb{E}(U^2) + \lambda p_{\{s\}} \mathbb{E}(W_s^2)}{2(1 - \lambda p_S \mathbb{E}(U) - \lambda p_{\{s\}} \mathbb{E}(W_s))} \right) \\ &= \frac{\lambda p_{\{s\}}}{1 - \lambda p_S q^{K-1} \mathbb{E}(Z)} \left(\mathbb{E}(Y)/\mu_s + \frac{\lambda p_S q^{K-2} \mathbb{E}(Z^2) + \lambda p_{\{s\}} \frac{1}{q} \mathbb{E}(Y^2)/\mu_s^2}{2(1 - \lambda p_S q^{K-1} \mathbb{E}(Z) - \lambda p_{\{s\}} \mathbb{E}(Y)/\mu_s)} \right) \end{aligned}$$

for $s = 1, \ldots, K$. We therefore obtain,

$$\begin{aligned} \mathbb{E}(N^{\mathrm{MRF}}) &= \sum_{s \in S} \mathbb{E}(N_{\{s\}}^{\mathrm{MRF}}) + \mathbb{E}(N_S^{\mathrm{MRF}}) \\ &= \frac{1}{q} \sum_{s=1}^{K} \frac{\lambda^2 p_{\{s\}}^2 \mathbb{E}(Y^2)/\mu_s^2}{2(1 - \lambda p_{\{s\}} \mathbb{E}(Y)/\mu_s)} + o(1/q) \end{aligned} \tag{5}$$

Under LRF, class s receives preemptive priority, so that their mean queue length is given by

$$\begin{aligned} \mathbb{E}(N_{\{s\}}^{\mathrm{LRF}}) &= \lambda p_{\{s\}} \left(\mathbb{E}(W_s) + \frac{\lambda p_{\{s\}} (W_s^2)}{2(1 - \lambda p_{\{s\}} \mathbb{E}(W_s))} \right) \\ &= \lambda p_{\{s\}} \left(\frac{\mathbb{E}(Y)}{\mu_s} + \frac{1}{q} \frac{\lambda p_{\{s\}} \mathbb{E}(Y^2)/\mu_s^2}{2(1 - \lambda p_{\{s\}} \mathbb{E}(Y)/\mu_s)} \right). \end{aligned}$$

Hence,

$$\sum_{s\in S}\mathbb{E}(N^{\mathrm{LRF}}_{\{s\}}) = \frac{1}{q}\sum_{s=1}^{K}\frac{\lambda^2 p^2_{\{s\}}\mathbb{E}(Y^2)/\mu_s^2}{2(1-\lambda p_{\{s\}}\mathbb{E}(Y)/\mu_s)} + o(1/q),$$

where we note that the term multiplied by $1/q$ coincides with that of $\mathbb{E}(N^{\mathrm{MRF}})$. In order to conclude the proof, it remains to prove that $\mathbb{E}(N^{\mathrm{LRF}}_S) = C/q + o(1/q)$, with $C > 0$.

Class-S jobs see a single-server queue where the capacity depends on the number of classes of dedicated traffic that are present. For example, when a class-S job enters service and there are n classes of dedicated traffic present, it can be served in $K - n$ servers. This dependence on the dedicated traffic makes it infeasible to obtain a closed-form expression for the mean number of class-S jobs under LRF. We therefore consider the following lower bound on $N^{LRF}_S(t)$. Consider a single-server system with capacity 1 and only class-S jobs. The server is on vacation whenever in the original LRF system all servers are working on dedicated traffic. The service time of a class-S job is distributed according to Z/q with probability q^K, and zero otherwise. This single-server system is a lower bound on $N^{LRF}_S(t)$, since (i) whenever class S is served in the original system, it is also served in the lower-bound system, and (ii) the generic service time of a job in the lower bound system is less than or equal to the generic service time in the original system in the server in which a copy of this job finishes service.

In the lower-bound system, we bound the mean sojourn time for this single-server queue with server interruptions following the "tagged job" technique as in [17]. That is, we imagine that a job, called the tagged job, enters the system and consider all the work that must be completed before it can leave. This is larger than the time it takes until class-S can be served. This time is either zero (in case the tagged job arrived outside a vacation) or is distributed as V, where the random variable V denotes the time left of the vacation. Hence, $\mathbb{E}(N^{LB}_S) \geq \lambda p_S \mathbb{P}(\text{the tagged job finds the server interrupted})\mathbb{E}(V)$.

We first compute the probability that the tagged job finds the server interrupted. The LB system is interrupted as long as all servers are busy in the original system. For the original system, the long-run proportion of time server s is busy serving class-$\{s\}$ jobs is $\lambda p_{\{s\}}\mathbb{E}(W_s) = \lambda p_{\{s\}}\mathbb{E}(Y)/\mu_s$. Because classes $1,\ldots,K$ are served independently, the long-run proportion of time that all servers are busy serving these classes is $\tilde{p} := \Pi_{s\in S}\lambda p_{\{s\}}\mathbb{E}(W_s) = \Pi_{s\in S}\lambda p_{\{s\}}\mathbb{E}(Y)/\mu_s$.

The interruption of the LB system is completed as soon as in the original system there is a server that completes all its class-$\{s\}$ jobs. The time that each server needs to complete its work of class $\{s\}$, T_s/μ_s is lower bounded by the time that the server needs to serve the job in service. Because the service times of jobs with a strictly positive service time are NWU, T_s is stochastically larger than Y_s/q. Hence,

$$\mathbb{E}(N^{LRF}_S) \geq \lambda p_S \tilde{p}\mathbb{E}(V) \geq \lambda p_S\tilde{p}\mathbb{E}(\min_s\{T_s/\mu_s\}) \geq \frac{\lambda p_S\tilde{p}}{q}\mathbb{E}(\min_s\{Y_s/\mu_s\}),$$

that is, $\mathbb{E}(N^{LRF}_S)$ is lower bounded by some strictly positive constant divided by q. □

Proof of Proposition 5: We assume that at time t under policy π, class $\{1,2\}$ has highest priority at server s, and that π starts serving a job, call it job b, on server s, where a copy of job b has already received some service on the other server $\hat{s}$, and there is another class-$\{1,2\}$ job, call it job a, that has not yet received any service. Let π' serve job a instead of job b on server s whenever π serves job b on server s and otherwise agree with π until job b completes under π, at time τ say. We couple all arrival and service times, including the service time on server s starting at time t. If job b completes on server $\hat{s}$ at time τ under π then it also completes on server $\hat{s}$ under π'. Due to the NBU assumption, we can couple the remaining service time of job a on server s under policy π' such that it is with probability 1 smaller than the service time of job a on server s under policy

π. Now, letting π' agree with π except for possibly idling when π is serving job a but job a has completed under π', we again have $\{N'(t)\}_{t\geq 0} \leq_{st} \{N(t)\}_{t\geq 0}$. If job b completes under π on server s at time τ, then job a completes under π' on server s. Let us relabel job b under π' as job a. Then the systems under π and π' are in the same state, except that job a has received some service under π' and not under π. Arguing as before, $\{N'(t)\}_{t\geq 0} \leq_{st} \{N(t)\}_{t\geq 0}$.

Finally, the construction of π' is similar if π idles server s when there is a job of class $\{s\}$. $\square$

We now explain why Proposition 5 cannot be generalized to more than two servers: At time t, there is a job in server 1 that has received some service, call this job b. We assume that under policy π, server 2 serves a fresh copy of job b, and that under policy π', server 2 serves a copy of a job that has received no service yet, call this job a. The proof of Proposition 5 relies on the fact that we can find a coupling argument where the next job that departs is job b under both π and π'. However, this coupling argument does not hold when there are more than two servers. At this particular state, a departure from a server 3 can lead to a state where job a leaves both systems π and π', but job b does not. Moreover, job b has received more service under policy π than under policy π', which is the opposite of what the coupling argument needs for the result to hold.

Proof of Corollary 6: We couple the service times of the classes $\{s\}$, for $s = 1, 2$. Since the second-level policy is non-preemptive, FCFS and ROS will be equivalent for class-$\{s\}$ jobs, for $s = 1, 2$.

Assume that at time t, jobs of class $\{1, 2\}$ have priority in server s. Under π we let the next job of this class in service. We call this job job b, and it might or might not have another copy in service in server $\hat{s}$. Under policy π' we let server s serve a class-$\{1, 2\}$ job chosen uniformly at random among the ones present in the queue, call it job a. We can have three main different scenarios:

$i)$ Job a and job b are the same job, that is, the copy that enters service in server s under policy π' is a copy of job b. Then, we couple the service times of both jobs, and π and π' will be in the same state from time t on.

$ii)$ Job a and job b are different jobs, and there is a copy of job b in server $\hat{s}$ that has received some service. By repeating the second part of the argument in Proposition 5, we have that $\{N'(t)\} \leq \{N(t)\}$.

$iii)$ Job a and job b are different jobs, and job b did not receive any service in server $\hat{s}$. In that case, the job in server $\hat{s}$ is either of class $\{s\}$ or a copy of another class-$\{1, 2\}$ job. In the first case, we rename the job a in π' to be job b and let π' agree with π, as we do in the second part of the argument in Proposition 5. If there is a copy of another class-$\{1, 2\}$ job in server $\hat{s}$, this also happens under π'. We couple the service times of the copies in server s and $\hat{s}$ under π and π'. Here we are again in the same scenario as in the second part of the argument in Proposition 5. Hence, $\{N'(t)\} \leq \{N(t)\}$. $\square$

Proof of Proposition 7: We first consider $\mathbb{E}(N^{MRF-FCFS})$. Because jobs have identical copies, we know that class-S jobs will complete service in the server with highest capacity, that is, $\mathbb{E}(X_S) = \min_{s\in S}\{1/\mu_s\}$, where X_c is the service time of a type-c job. Class S sees a single server with capacity $\max_{s\in S}\{\mu_s\}$, and class $\{i\}$ a priority queue where class S receives priority over class $\{i\}$. We assume, without loss of generality, that $\mu_1 = \max_{s\in S}\{\mu_s\}$. Using [17], we then obtain

$$\begin{aligned}\mathbb{E}(N^{MRF}) &= \lambda p_S \left(\frac{1}{\mu_1} + \frac{\lambda p_S \frac{1}{2\mu_1}}{1-\lambda p_S \frac{1}{\mu_1}}\right) \\ &\quad + \sum_{i=1}^{K} \frac{\lambda p_{\{i\}}}{1-\lambda p_S \frac{1}{\mu_1}} \left(\frac{1}{\mu_i} + \frac{\lambda p_{\{i\}} \frac{1}{2\mu_i} + \lambda p_S \frac{1}{2\mu_1}}{1-\lambda p_S \frac{1}{\mu_1} - \lambda p_{\{i\}} \frac{1}{\mu_i}}\right). \qquad (6)\end{aligned}$$

For the LRF-FCFS policy, we do not have a closed-form expression for the mean number of jobs. We therefore construct an upper-bound (UB) system under the LRF-FCFS policy where a class-S job departs only when the copy in server 1 is served. One can easily verify that this system upper bounds the original system. The mean number of jobs under the upper-bound system can be easily calculated, since now class $\{i\}$ sees a single server queue and class S a priority queue where class 1 is given priority. Using [17], we then obtain

$$\begin{aligned}\mathbb{E}(N^{UB}) &= \sum_{i=1}^{K} \lambda p_{\{i\}} \left(\frac{1}{\mu_i} + \frac{\lambda p_{\{i\}} \frac{1}{2\mu_i}}{1 - \lambda p_{\{i\}} \frac{1}{\mu_i}} \right) \\ &\quad + \frac{\lambda p_S}{1 - \lambda p_{\{1\}} \frac{1}{\mu_1}} \left(\frac{1}{\mu_1} + \frac{\lambda p_{\{1\}} \frac{1}{\mu_1} + \lambda p_S \frac{1}{\mu_1}}{2(1 - \lambda p_{\{1\}} \frac{1}{\mu_1} - \lambda p_S \frac{1}{\mu_1})} \right). \end{aligned} \tag{7}$$

We can now compare $\mathbb{E}(N^{MRF})$ with $\mathbb{E}(N^{UB})$. We introduce some notation to make the computations easier:

$$a = 1 - \lambda p_S \frac{1}{\mu_1}, \qquad b = 1 - \lambda p_{\{1\}} \frac{1}{\mu_1}, \qquad c = 1 - \lambda p_S \frac{1}{\mu_1} - \lambda p_{\{1\}} \frac{1}{\mu_1},$$

$$\text{and for } i = 1, \ldots, K, \qquad d_i = 1 - \lambda p_S \frac{1}{\mu_1} - \lambda p_{\{i\}} \frac{1}{\mu_i}, \qquad e_i = 1 - \lambda p_{\{i\}} \frac{1}{\mu_i}.$$

Let us start by comparing the performance on server $i¿1$, that is, the terms multiplied by $\lambda p_{\{i\}}$, $i > 1$, in Eq. (6) and Eq. (7).

$$\begin{aligned}&\sum_{i=2}^{K} [\mathbb{E}(N_i^{MRF}) - \mathbb{E}(N_i^{UB})] \\ &= \sum_{i=2}^{K} \left[\frac{\lambda p_{\{i\}}}{a} \left(\frac{1}{\mu_i} + \frac{\lambda p_{\{i\}} \frac{1}{\mu_i} + \lambda p_S \frac{1}{\mu_1}}{2 d_i} \right) - \lambda p_{\{i\}} \left(\frac{1}{\mu_i} + \frac{\lambda p_{\{i\}} \frac{1}{\mu_i}}{2 e_i} \right) \right] \\ &= \sum_{i=2}^{K} \left[\frac{\lambda p_{\{i\}} \frac{1}{\mu_i} (\lambda p_S \frac{1}{\mu_1})}{a} + \frac{\lambda^2 p_{\{i\}}^2 \frac{1}{\mu_i}}{2 a d_i e_i} \left(\lambda p_S \frac{1}{\mu_1} (d_i + 1) \right) + \frac{\lambda^2 p_{\{i\}} p_S \frac{1}{\mu_1}}{2 a d_i} \right] \\ &\geq \sum_{i=2}^{K} \left[\frac{\lambda^2 p_{\{i\}} p_S}{a \mu_1^2} + \frac{\lambda^2 p_{\{i\}} p_S}{2 a d_i e_i \mu_1} \left(\lambda p_{\{i\}} \frac{1}{\mu_1} (d_i + 1) + e_i \right) \right] > 0. \end{aligned} \tag{8}$$

The last inequality holds since we have assumed that $\mu_1 = \max_{s \in S} \{\mu_s\}$. This difference is strictly positive for any parameter values. Let us compute the difference now in server 1, that is, the terms in (6) and (7) multiplied by $\lambda p_{\{1\}}$ and λp_S.

$$\begin{aligned}&\mathbb{E}(N_1^{MRF}) - \mathbb{E}(N_1^{UB}) \\ &= \frac{\lambda p_{\{1\}}}{a \mu_1} \left(1 + \frac{\lambda p_{\{1\}} + \lambda p_S}{2c} \right) - \frac{\lambda p_{\{1\}}}{\mu_1} \left(1 + \frac{\lambda p_{\{1\}}}{2b} \right) \\ &\quad + \frac{\lambda p_S}{\mu_1} \left(1 + \frac{\lambda p_S}{2a} \right) - \frac{\lambda p_S}{b \mu_1} \left(1 + \frac{\lambda p_{\{1\}} + \lambda p_S}{2c} \right) \\ &= \frac{\lambda^2 p_{\{1\}} p_S}{\mu_1^2 a} + \frac{\lambda^2 p_{\{1\}}^2}{2abc\mu_1^2} (\lambda p_S (c+1)) + \frac{\lambda^2 p_{\{1\}} p_S}{2ac\mu_1} \\ &\quad - \frac{\lambda^2 p_{\{1\}} p_S}{b \mu_1^2} - \frac{\lambda^2 p_S^2}{2abc\mu_1^2} \left(\lambda p_{\{1\}} (c+1) \right) - \frac{\lambda^2 p_{\{1\}} p_S}{2bc\mu_1} \\ &= \left(p_S - p_{\{1\}} \right) \left(\frac{\lambda^3 p_{\{1\}} p_S}{ab\mu_1^3} + \frac{\lambda^3 p_{\{1\}} p_S (1-c)}{2abc\mu_1^2} + \frac{\lambda^3 p_{\{1\}} p_S}{2abc\mu_1^2} \right). \end{aligned} \tag{9}$$

We note that this is nonnegative if and only if $p_S \geq p_{\{1\}}$.

We can obtain a more accurate result by summing the terms multiplied by λ^3 in Equation (8) restricted to class $\{i\}$ and the last term in Equation (9). We have then,

$$\frac{\lambda^3 p_S}{2a\mu_1^2}\left(\frac{p_{\{i\}}^2(d_i+1)}{d_i e_i}+\frac{p_{\{1\}}(p_S-p_{\{1\}})}{cb}\right).$$

We note that a, b, c, d_i and e_i are positive. Hence, the first multiplying term is positive. The latter implies that the whole expression is positive if and only if

$$\frac{p_{\{i\}}^2(d_i+1)}{d_i e_i}+\frac{p_{\{1\}}(p_S-p_{\{1\}})}{cb} > 0 \Longleftrightarrow$$

$$\frac{p_{\{i\}}^2(2-\frac{\lambda p_S}{\mu_1}-\frac{\lambda p_{\{i\}}}{\mu_i})}{(1-\frac{\lambda p_S}{\mu_1}-\frac{\lambda p_{\{i\}}}{\mu_i})(1-\frac{\lambda p_{\{i\}}}{\mu_i})}+\frac{p_{\{1\}}(p_S-p_{\{1\}})}{(1-\frac{\lambda p_{\{1\}}}{\mu_1})(1-\frac{\lambda p_S}{\mu_1}-\frac{\lambda p_{\{1\}}}{\mu_1})} > 0$$

We define $\rho_j = p_{\{j\}}/\mu_j$ for $j = 1, i$ and $\rho_S = p_S/\mu_1$.

$$\Longleftrightarrow \lambda^2(\rho_S+\rho_i)[p_{\{i\}}^2(\rho_S+\rho_i)+p_{\{1\}}(p_S-p_{\{1\}})\rho_i]$$

$$-\lambda[p_{\{i\}}^2(2\rho_1+3\rho_S+\rho_i)+p_{\{1\}}(p_S-p_{\{1\}})(2\rho_i+\rho_S)]+2p_{\{i\}}^2+p_{\{1\}}(p_S-p_{\{1\}})>0.$$

For ease of notation, let

$$\begin{aligned}
\tilde{a}_i &= (\rho_S+\rho_i)[p_{\{i\}}^2(\rho_S+\rho_i)+p_{\{1\}}(p_S-p_{\{1\}})\rho_i],\\
\tilde{b}_i &= p_{\{i\}}^2(2\rho_1+3\rho_S+\rho_i)+p_{\{1\}}(p_S-p_{\{1\}})(2\rho_i+\rho_S),\\
\tilde{c}_i &= 2p_{\{i\}}^2+p_{\{1\}}(p_S-p_{\{1\}}).
\end{aligned}$$

Hence, if λ is either smaller than $\lambda_{0,i}$ or larger than $\lambda^{0,i}$, then the mean number of jobs under the MRF system is strictly larger than that under UB (and hence under LRF), where

$$\lambda_{0,i}=\frac{\tilde{b}_i-\sqrt{\tilde{b}_i^2-4\tilde{a}_i\tilde{c}_i}}{2\tilde{a}_i}, \text{ and } \lambda^{0,i}=\frac{\tilde{b}_i+\sqrt{\tilde{b}_i^2-4\tilde{a}_i\tilde{c}_i}}{2\tilde{a}_i}, \tag{10}$$

Thus, we define

$$\lambda_0=\min_{i=2}^{K}\{\lambda_{0,i}\}=\min_{i=2}^{K}\left\{\frac{\tilde{b}_i-\sqrt{\tilde{b}_i^2-4\tilde{a}_i\tilde{c}_i}}{2\tilde{a}_i}\right\}. \tag{11}$$

In particularly, if the arrival rate λ is smaller than λ_0, then the mean number of jobs under the MRF system is strictly larger than that under UB (and hence under LRF). □

Proof of Proposition 11: We actually show a stronger result: we prove that the system with i.i.d. copies is optimal when for each job, we can choose either independent or identical copies.

We consider a policy π where servers are required to follow Π_1-FCFS and idling is allowed. Under policy π, whenever a server starts serving a first copy of a job, that server determines whether the service times of the copies of that job are identical or i.i.d.. This decision is independent of the history of the process. We show that for this policy π, if at time t it idles or schedules identical copies for a job in service for the first time, we can construct a policy π' with a coupled sample path, such that at time t π' does not idle or samples i.i.d. copies, and such that

$\{N'(s)\}_{s\geq 0} \leq \{N(s)\}_{s\geq 0}$ with probability 1, where N (N') denotes the number of jobs in the system under policy π (π'). The result follows by starting at time 0 and repeating the argument each time a policy deviates from the policy π, until we have the non-idling Π_1-FCFS policy with i.i.d. copies.

One can easily verify that the non-idling Π_1-FCFS policy is better than idling Π_1-FCFS, by following the steps in the second part of the proof of Proposition 3.

Therefore let us assume that π never idles but that at some time t, π starts serving the first copy of some job, call it job 1, and π chooses identical copies for that job. We let π' agree with π before time t and let it choose i.i.d. copies for job 1. We denote by τ the time that job 1 completes service under π, say at server 1. Because the copies are identical, server 1 has done the most work on job 1 between times t and τ. We couple the service time of job 1 on server 1 under π' to that under π. Then, π' samples i.i.d. copies for the rest of the copies of job 1. We let all other service times and all arrival times be coupled under both policies. We let π' agree with π until job 1 departs under π', at time τ'. From our coupling, $\tau' \leq \tau$. Let π' idle any server that serves job 1 under π between times τ' and τ and let it otherwise agree with π from time τ' on. From our argument above, a policy that agrees with π' but does not idle will have even earlier departures than π'. The argument can be repeated, each time reducing the number-in-system process, until we have all i.i.d. copies and no idling. □

B: Proofs of Section 4

Proof of Proposition 15 and Proposition 16

In order to prove both propositions, we analyze the fluid-scaled system. We recall that the redundancy structure is nested and that $N_c(t)$ denotes the number of class-c jobs at time t. For $r > 0$, we denote by $N_c^r(t)$ the system where the initial state satisfies $N_c^r(0) = rn_c(0)$, for all $c \in \mathcal{C}$. We write the fluid-scaled number of jobs per class by using standard arguments, see [9],

$$\frac{N_c^r(rt)}{r} = n_c(0) + \frac{1}{r}\tilde{A}_c(rt) - \frac{1}{r}\tilde{S}_c(T_c^r(rt)), \tag{12}$$

where $\tilde{A}_c(t)$ and $\tilde{S}_s(t)$ are independent Poisson processes having rates λp_c and 1, respectively. $T_c^r(t)$ is the cumulative amount of capacity spent in serving class-c jobs, which strongly depends upon the correlation structure among the copies, that is,

$$T_c^r(t) = g((T_{s,c}^r(t))_{s\in c}),$$

where $T_{s,c}^r(t)$ is the cumulative amount of capacity spent on serving class-c jobs in server $s \in c$ during the time interval $(0, t]$ and g is characterized by the correlation structure of the copies. We note that when copies are i.i.d. $T_c^r(t) = \sum_{s\in c} T_{s,c}^r(t)$ and when copies are identical, $T_c^r(t) = \max_{s\in c}\{T_{s,c}^r(t)\}$.

In the following result, we obtain the general characterization of a fluid limit. The existence of fluid limits can be proved following the same steps as in [9], so its proof is omitted.

Lemma 18. *For almost all sample paths ω and sequence $r_k \to \infty$, there exists a subsequence $r_{k_j} \to \infty$ such that for all $c \in \mathcal{C}$ and $t \geq 0$,*

$$\lim_{j\to\infty} \frac{N_c^{r_{k_j}}(r_{k_j}t)}{r_{k_j}} = n_c(t) \text{ } u.o.c^1 \text{ } \textit{and} \lim_{j\to\infty} \frac{T_c^{r_{k_j}}(r_{k_j}t)}{r_{k_j}} = \tau_c(t) \text{ } u.o.c., \tag{13}$$

with $(n_c(\cdot), \tau_c(\cdot))$ continuous functions. In addition,

$$n_c(t) = n_c(0) + \lambda p_c t - \tau_c(t), \tag{14}$$

where $n_c(t) \geq 0$, $\tau_c(0) = 0$, $\tau_c(t) \leq t \max_{s \in c}\{\mu_s\}$, and $\tau_c(\cdot)$ are non-decreasing and Lipschitz continuous functions for all $c \in \mathcal{C}$.

In order to provide the characterization of the fluid limit, we first introduce some notation. Let us group the classes with respect to their number of copies. We denote by $\mathcal{L}_i$ the set of classes with i copies, that is, for $i = 2, \ldots, |C|$,

$$\mathcal{L}_i = \{c \in \mathcal{C} : |c| = i\}.$$

From the nested structure of the system, we note that for each $c \in \mathcal{L}_i$ and $\tilde{c} \in \mathcal{L}_j$ with $j < i$, either $\tilde{c} \subset c$ or $\tilde{c} \cap c = \emptyset$. For all $c \in \mathcal{L}_i$, let us denote by $\mathcal{L}_i(c) = \{\tilde{c} \subset c\}$ the job classes that are subsumed in class c, for $i = 1, \ldots, |C|$.

We denote by $P_s(\vec{N})$, the probability that, given $\vec{N}(0) = \vec{N}$, at time $t = 0$ a given server s is serving a copy that is not in service in any other server. Then, the following lemma is true.

Lemma 19. *Consider a redundancy system with a nested topology and where servers implement Π_1-ROS. For any server $s \in S$ and $\vec{N}^r(0) = r\vec{n}^r$, such that $\lim_{r\to\infty} \sum_{c \in \mathcal{C}(s)} r n_c^r > 0$, then*

$$\lim_{r\to\infty} P_s(r\vec{n}^r) = 1. \tag{15}$$

Proof: Assume at time 0, server s idles and that we are in state $\vec{N}^r(0) = \vec{N}$. At this moment, under Π_1-ROS server s can only serve a single class in the system, say class c. Let us consider servers $l \in c$ that are serving a class-c job at time $t = 0$, say servers $S(c)$. We denote by $-\tilde{T}_l^r < 0$ the time at which server l started serving a new class-c job, regardless of whether another job departed from the server, or the class-c job preempted a copy with lower priority in that server. We note that $\frac{N_c(-\tilde{T}_l^r)-1}{N_c(-\tilde{T}_l^r)}$ is the probability that server l is *not* serving the copy of the same job that is now in service in server s. Hence,

$$P_s(\vec{N}) = \Pi_{l \in S(c), l \neq s} \frac{N_c^r(-\tilde{T}_l^r) - 1}{N_c^r(-\tilde{T}_l^r)}. \tag{16}$$

We set $\vec{N}(0) = r\vec{n}^r$. Since the transition rates μ_s and λ are of order $O(1)$, it follows directly that $\tilde{T}_s^r$ and $\vec{N}(-\tilde{T}_s^r) - \vec{N}(0)$ are of order $O(1)$ as well, so that

$$\lim_{r\to\infty} \frac{N_c^r(-\tilde{T}_l^r) - 1}{N_c^r(-\tilde{T}_l^r)} = \lim_{r\to\infty} \frac{N_c^r(0) - 1}{N_c^r(0)} = 1. \tag{17}$$

It hence follows from (16) that $\lim_{r\to\infty} P_s(r\vec{n}^r) = 1$. □

Let us characterize the instantaneous departure rate of a class-$\tilde{c}$ job. Let us denote by $\mathcal{C}_{\tilde{c}}$ the classes that are a subsumed in class $\tilde{c}$. That is,

$$\mathcal{C}_{\tilde{c}} = \{c \in \mathcal{C} \ : \ c \subseteq \tilde{c}\}.$$

Note that if $\tilde{c} \in \mathcal{L}_i$, then $\mathcal{C}_{\tilde{c}} = \mathcal{L}_i(\tilde{c})$.

Lemma 20. *Assume that jobs have exponential service times and*

- *if copies are i.i.d. copies, assume LRF-Π_2, with LRF non-preemptive and Π_2 non-idling,*
- *if copies follow some general correlation structure, assume LRF-ROS.*

For each class $\tilde{c} \in \mathcal{C}$, the fluid limit $\sum_{c\in\mathcal{C}_{\tilde{c}}} n_c(t)$ satisfies the following:

$$\frac{\mathrm{d}\sum_{c\in\mathcal{C}_{\tilde{c}}} n_c(t)}{\mathrm{d}t} = \sum_{c\in\mathcal{C}_{\tilde{c}}} \lambda p_c - \sum_{s\in\tilde{c}} \mu_s, \;\; \textit{if} \;\; n_{\tilde{c}}(t) > 0.$$

Proof: We first consider a general correlation structure among the copies and LRF-ROS. When starting in state $\vec{N}(0) = r\vec{n}^r$, the drift function is

$$\begin{aligned}\tilde{f}(r\sum_{c\in\mathcal{C}_{\tilde{c}}} n_c^r) \;\; &= \;\; \sum_{c\in\mathcal{C}_{\tilde{c}}} \lambda p_c \\ &\quad - \sum_{s\in\tilde{c}} \mu_s \left(\prod_{l\in\tilde{c}} P_l(r\vec{n}^r)\right) - g_{\tilde{c}}(r\vec{n}^r)(1 - \prod_{l\in\tilde{c}} P_l(r\vec{n}^r)) \end{aligned} \tag{18}$$

for $n_{\tilde{c}} > 0$, where $g_{\tilde{c}}$ is a function that captures the instantaneous departure rate of the system when more than one copy (with any correlation structure) of the same job is in service, and note that $g_{\tilde{c}} = O(1)$ as $r \to \infty$. Assuming that $n_{\tilde{c}} > 0$, due to LRF and the nested structure, each server $s \in \tilde{c}$ serve jobs of a single class c, with $c \subseteq \tilde{c}$ which is the class with least redundant copies in that server s. We note that the first departure term in Eq. (18) represents departures from servers $s \in \tilde{c}$ of class-c jobs $c \in \tilde{c}$, who were served in one unique server. Since these jobs have no other copies in service, the total departure rate from the servers in $\tilde{c}$ equals $\sum_{s\in\tilde{c}} \mu_s$. The second departure term in (18) represents departures due to a class-c job that is being served in more than one server simultaneously.

Therefore, from equation 18 together with Lemma 19, we obtain

$$\lim_{r\to\infty} \tilde{f}(r\sum_{c\in\mathcal{C}_{\tilde{c}}} n_c^r) = \lambda \sum_{c\in\mathcal{C}_{\tilde{c}}} p_c - \sum_{s\in\tilde{c}} \mu_s. \tag{19}$$

Now assume copies are i.i.d. copies and servers implement LRF-Π_2 with LRF non-preemptive and Π_2 non-idling. Under these assumptions, the drift function is given by

$$\tilde{f}(r\sum_{c\in\mathcal{C}_{\tilde{c}}} n_c^r) = \lambda \sum_{c\in\mathcal{C}_{\tilde{c}}} p_c - \sum_{s\in\tilde{c}} \mu_s, \text{ when } n_{\tilde{c}} > 0. \tag{20}$$

This can be seen as follows. When there are class-c jobs present, $c \in \tilde{c}$, each server in class $\tilde{c}$ is working on such a job, possibly the same job. Because of the i.i.d. copies assumption, the departure rate of these jobs is simply the sum of all the capacities. □

In order to prove Proposition 15 and Proposition 16, we introduce the redundancy degree per class. Let us assume w.l.o.g. that $\mathcal{L}_1$ is non-empty. For each class $c \in \mathcal{L}_1$, the fluid limit $n_c(t)$ is given by the following due to Lemma 20:

$$\frac{\mathrm{d}n_c(t)}{\mathrm{d}t} = \lambda p_c - \sum_{s\in c} \mu_s, \;\; \text{if} \;\; n_c(t) > 0.$$

We note that $\lambda p_c - \sum_{s\in c} \mu_s < 0$, by assumption. This coincides with the fluid limit of an $M/M/1$ system with arrival rate λp_c and server capacity $\sum_{s\in c} \mu_s$. Hence, for all $c \in \mathcal{L}_1$, $n_c(t)$ reaches zero in finite time, say at time T_1, and stays zero.

The proof follows now by induction. Assume that there is a time T_j, such that $n_c(t) = 0$ for $t > T_j$, for all $c \in \mathcal{L}_i$ and $i \leq j$. In the following we show that for $\mathcal{L}_{j+1}$, there is a $T_{j+1} > T_j$, such that $n_c(t) = 0$ for $t > T_{j+1}$ and for all $c \in \mathcal{L}_{j+1}$.

For a class $c \in \mathcal{L}_{j+1}$, the fluid drift of $\sum_{\tilde{c}\in\mathcal{L}_{j+1}(c)} n_{\tilde{c}}(t)$ is given by the following due to Lemma 20:

$$\frac{\mathrm{d}\sum_{\tilde{c}\in\mathcal{L}_{j+1}(c)} n_{\tilde{c}}(t)}{\mathrm{d}t} = \sum_{\tilde{c}\in\mathcal{L}_{j+1}(c)} \lambda p_{\tilde{c}} - \sum_{s\in c} \mu_s, \text{ if } n_c(t) > 0. \tag{21}$$

From the induction hypothesis, we note that there exits time T_j such that $n_{\tilde{c}}(t) = 0$ for all $\tilde{c} \in \mathcal{L}_i$ with $i \leq j$. Hence, $\frac{\mathrm{d}\sum_{\tilde{c}\in\mathcal{L}_{j+1}(c)} n_{\tilde{c}}(t)}{\mathrm{d}t} = \frac{\mathrm{d}n_c(t)}{\mathrm{d}t}$, for $t \geq T_j$. Together with Eq. 21,

$$\frac{\mathrm{d}n_c(t)}{\mathrm{d}t} = \sum_{\tilde{c}\in\mathcal{L}_{j+1}(c)} \lambda p_{\tilde{c}} - \sum_{s\in c} \mu_s, \text{ if } n_c(t) > 0,$$

for all $t \geq T_j$. We note that $\lambda \sum_{\tilde{c}\in\mathcal{L}_{j+1}(c)} p_{\tilde{c}} - \sum_{s\in c} \mu_s < 0$, by assumption. This coincides with the fluid limit of an $M/M/1$ system with arrival rate $\lambda \sum_{\tilde{c}\in\mathcal{L}_{j+1}(c)} p_{\tilde{c}}$ and server capacity $\sum_{s\in c} \mu_s$. Hence, for all $c \in \mathcal{L}_{j+1}$, $n_c(t)$ reaches zero in finite time, say at time T_{j+1}, and stays zero. Hence, there exists time $\tilde{T} > 0$ when the fluid process is empty. Together with [24], we conclude that the system is stable. □