The Power of d Choices for Redundancy

Kristen Gardner
Carnegie Mellon University
Pittsburgh, PA
ksgardne@cs.cmu.edu

Samuel Zbarsky
Carnegie Mellon University
Pittsburgh, PA
szbarsky@andrew.cmu.edu

Mor Harchol-Balter
Carnegie Mellon University
Pittsburgh, PA
harchol@cs.cmu.edu *

Alan Scheller-Wolf
Carnegie Mellon University
Pittsburgh, PA
awolf@andrew.cmu.edu

ABSTRACT

An increasingly prevalent technique for improving response time in queueing systems is the use of redundancy. In a system with redundant requests, each job that arrives to the system is copied and dispatched to multiple servers. As soon as the first copy completes service, the job is considered complete, and all remaining copies are deleted. A great deal of empirical work has demonstrated that redundancy can significantly reduce response time in systems ranging from Google's BigTable service to kidney transplant waitlists.

We propose a theoretical model of redundancy, the Redundancy-d system, in which each job sends redundant copies to d servers chosen uniformly at random. We derive the first exact expressions for mean response time in Redundancy-d systems with any finite number of servers. We also find asymptotically exact expressions for the distribution of response time as the number of servers approaches infinity.

1. INTRODUCTION

Redundancy – the idea of dispatching multiple copies of the same job and waiting for the first copy to complete service – is an important strategy for reducing response times in applications ranging from Google's BigTable service to kidney transplant waitlists.

Redundancy provides significant response time improvements because it exploits two sources of variability. First, queueing times across servers can be highly variable due to load from different applications. Redundant requests wait in the queue at multiple servers, so they experience the *minimum queueing time* across these servers. Second, the *same* job might see highly variable service times at different servers. For example, in computer systems applications such as web queries, external factors such as network interference, disk seek time, and background tasks can cause a query to be slowed down unpredictably. This slowdown dominates the computation time required for the query, which is inherently quite small. This causes the query's actual service time to be very long relative to its inherent size. For example, a web query can be slowed down by up to a factor of 27 due to unpredictable background load [4]. Sending redundant requests enables a job to receive the *minimum service time* across servers.

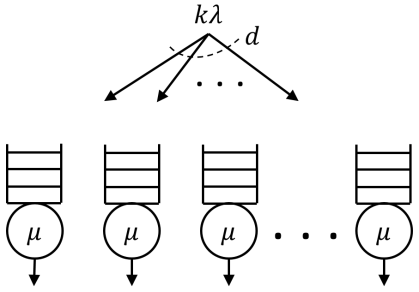

Figure 1: The Redundancy-d system consists of k servers, each providing independent exponential service times with rate μ. Jobs arrive to the system as a Poisson process with rate $k\lambda$. Each job sends copies to d servers chosen uniformly at random. A job is complete when its first copy completes service.

While it is clear that redundancy can lead to a significant reduction in response time, it is often difficult to determine how much redundancy is needed to obtain such improvements. Is sending only two copies enough to achieve most of the potential benefit? What is the additional benefit of increasing the number of copies?

We study these questions in a theoretical model called the Redundancy-d system (see Figure 1). The Redundancy-d system consists of k servers; each arriving job makes d copies of itself and dispatches these copies to d different servers chosen uniformly at random. The job is considered complete as soon as the first copy completes service.

Our *primary contribution* is providing the first exact analysis of response time in the Redundancy-d system. First, we derive exact closed-form expressions for mean response time as a function of the number of servers k and the number of

*This work was supported by NSF awards CMMI-1538204, CMMI-1334194, and CSR-1116282; by the Intel Science and Technology Center for Cloud Computing; and by a Google Faculty Research Award 2015/16.

SIGMETRICS '16 June 14-18, 2016, Antibes Juan-Les-Pins, France

ACM ISBN 978-1-4503-4266-7/16/06.

DOI: http://dx.doi.org/10.1145/2896377.2901497

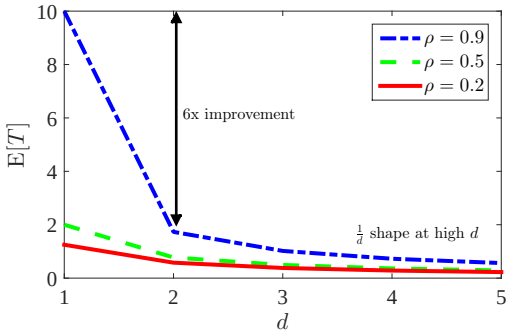

Figure 2: Mean response time $\mathrm{E}[T]$ as a function of d under low ($\rho = 0.2$, solid red line), medium ($\rho = 0.5$, dashed green line), and high ($\rho = 0.9$, dot-dashed blue line) load.

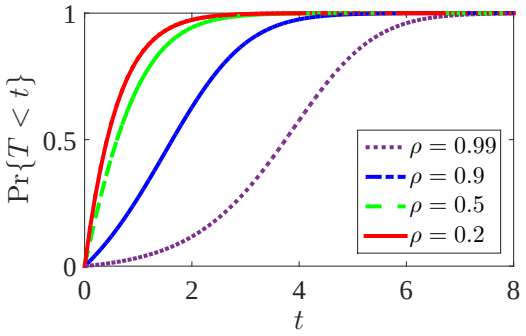

Figure 3: Probability that response time $T < t$ when $d = 2$ under low ($\rho = 0.2$, solid red line), medium ($\rho = 0.5$, dashed green line), high ($\rho = 0.9$, dot-dashed blue line), and very high ($\rho = 0.99$, dotted purple line) load.

copies per job, d, by modeling the system as a Markov chain. Second, we consider the system in the limit as the number of servers k approaches infinity. Under a standard asymptotic independence assumption, we derive an asymptotically exact expression for the distribution of response time. Our exact analysis allows us to quantify the magnitude of the benefit from increasing d.

2. MAIN RESULTS

Let $\rho = \frac{\lambda}{\mu}$ denote the system load. This is the total arrival rate to the system ($k\lambda$) divided by the maximum service rate of the system ($k\mu$). The system is stable as long as $\rho < 1$.

THEOREM 1. *The mean response time in the Redundancy-d system with k servers is*

$$\mathrm{E}[T] = \sum_{i=d}^{k} \frac{1}{k\mu \frac{\binom{k-1}{d-1}}{\binom{i-1}{d-1}} - k\lambda}. \quad (1)$$

We derive the result in Theorem 1 by modeling the system as a Markov chain. Following [1], our system state is a list of all jobs in the system in the order in which they arrived, where we track the d specific servers to which each job sent its copies. The general form of the limiting distribution of the state space is an immediate consequence of Theorem 1 in [1]. Unfortunately, knowing the limiting distribution's form does not tell us the normalizing constant, nor does it immediately yield results for mean number in system and mean response time. To find mean response time, we must first find $\Pr\{m \text{ jobs in system}\}$ by summing over all $\binom{k}{d}$ possible choices of servers for each queue position. We develop a novel state aggregation approach that uses recurrence relations to obtain the result given in Theorem 1 (see [2]).

THEOREM 2. *Assuming queues are d-wise asymptotically independent as $k \to \infty$, the response time in the Redundancy-d system with $d > 1$ has c.c.d.f.*

$$\Pr\{T > t\} = \left(\frac{1}{\rho + (1-\rho)e^{t\mu(d-1)}}\right)^{\frac{d}{d-1}}. \quad (2)$$

Theorem 2 relies on the assumption that queues are asymptotically independent as $k \to \infty$ (see [2] for definition); this is a standard assumption that has been shown to hold in several related systems (e.g., [3]). Our proof of Theorem 2 involves considering a tagged arrival to the Redundancy-d system, and asking why that arrival would have not yet departed by time t. There are two possibilities: either the job has a large size, or the job experiences a long queueing time. In the latter case, the long queueing time can again be attributed to the preceding arrival having a large size or a long queueing time. This recursive formulation leads to a system of differential equations, which we solve to obtain the result in Theorem 2.

Our exact analysis allows us to quantify the response time benefit obtained from increasing d. Figure 2 shows mean response time, $\mathrm{E}[T]$, in the Redundancy-d system as a function of d for low, medium, and high load. At all loads, $\mathrm{E}[T]$ decreases as d increases. The most significant improvement occurs between $d = 1$ and $d = 2$: $\mathrm{E}[T]$ decreases by up to a factor of 6. Figure 3 shows that this improvement is even bigger at the tail of response time; the 95th percentile of response time decreases by up to a factor of 8 when $\rho = 0.9$. From Theorem 2, we see that as d becomes large, mean response time scales in proportion to $\frac{1}{d}$, indicating that there is decreasing marginal benefit from further increasing d.

3. REFERENCES

[1] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, E. Hyytiä, and A. Scheller-Wolf. Reducing latency via redundant requests: Exact analysis. In *SIGMETRICS*, June 2015.

[2] K. Gardner, S. Zbarsky, M. Harchol-Balter, and A. Scheller-Wolf. Analyzing response time in the Redundancy-d system. Technical report, CMU-CS-15-141R, 2016.

[3] N. Vvedenskaya, R. Dobrushin, and F. Karpelevich. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Probl. Peredachi Inf.*, 32(1):20–34, 1996.

[4] Y. Xu, M. Bailey, B. Noble, and F. Jahanian. Small is better: Avoiding latency traps in virtualized data centers. In *Proceedings of the 4th annual Symposium on Cloud Computing*, page 7. ACM, 2013.