Contrasting Effects of Replication in Parallel Systems: From Overload to Underload and Back

Felix Poloczek
University of Warwick / TU Berlin

Florin Ciucu
University of Warwick

ABSTRACT

Task replication has recently been advocated as a practical solution to reduce latencies in parallel systems. In addition to several convincing empirical studies, analytical results have been provided, yet under some strong assumptions such as independent service times of the replicas, which may lend themselves to some contrasting and perhaps contriving behavior. For instance, under the independence assumption, an overloaded system can be stabilized by a replication factor, but can be sent back in overload through further replication. Motivated by the need to dispense with such common and restricting assumptions, which may cause unexpected behavior, we develop a unified and general theoretical framework to compute tight bounds on the distribution of response times in general replication systems. These results immediately lend themselves to the optimal number of replicas minimizing response time quantiles, depending on the parameters of the system (e.g., the degree of correlation amongst replicas).

1. INTRODUCTION

Given the late abundance of computing resources, a natural and yet very simple way to improve latencies is *replication*. In the context of a multi-server (parallel) system, the idea is merely to replicate a task into multiple copies/replicas, and to execute each replica on a different server. By leveraging the statistical variability of the servers themselves, as execution platforms, it is expected that some replicas would finish much faster than others. The key gain of executing multiple replicas is not to reduce the average latency, but rather the latency tail which is recognized as critically important for ensuring a consistently fluid/natural responsiveness of systems.

While the idea of using redundant requests is not new, as it has been used to demonstrate significant speedups in parallel programs [3], it has become very attractive with its implementation in the MapReduce framework through the so-called 'backup-tasks' [1]. Thereafter there has been a surge of high-quality empirical work which has convincingly demonstrated the benefits of using redundancy for significant latency improvement.

Such empirical work has been complemented by several excellent analytical studies, which have provided fundamental insight into the benefits of replication. Constrained by analytical tractability, most of these works make several strong assumptions: not only the arrivals are Poisson and the service times are exponentially distributed, but the service times of the replicas plus the corresponding original tasks are statistically independent [2]. By challenging these assumptions, we first provide some elementary analytical arguments, that the benefits of replication are highly dependent on both the distributional and correlation structures of the service times. Second, we develop a general analytical framework to compute stochastic bounds on the response time distributions in replication systems. In particular, our framework covers scenarios with Markovian arrivals, general service time distributions, and a correlation model amongst the original and replicated tasks.

2. ELEMENTARY INSIGHTS

We consider a parallel system with K homogeneous servers with identical speeds. A stream of tasks arrives at a dispatcher according to some stationary and ergodic point process; the interarrival times are denoted by t_i with mean $E[t_1] = \frac{1}{\lambda}$.

Upon its arrival, job i is replicated to $k \leq K$ servers where they are processed with service times $x_{i,1}, \ldots, x_{i,k}$, respectively. We first assume the family of service times $\{x_{i,j} \mid i \geq 1,\ 1 \leq j \leq k\}$ are i.i.d. and drawn from some general distribution subject to a finite moment generating function; the average is set to $E[x_1] = \frac{1}{\mu}$.

A necessary and sufficient condition for stability in a scenario with replication factor $0 \leq k \leq K$ is

$$\mathbb{E}[\min\{x_1, \ldots, x_k\}] < \frac{K}{k}\mathbb{E}[t_1]\ . \tag{1}$$

Denoting the CCDF of x_i by $f(x) := \mathbb{P}(x_1 \geq x)$, we observe from the previous stability conditions that the 'best' replication-factor k is

$$\operatorname*{argmin}_k k \int f^k(x)dx\ . \tag{2}$$

Depending on the distribution of the x_i, each of the replication strategies 1) No-Replication (i.e., $k = 1$), 2) Full-Replication (i.e., $k = K$), and 3) Partial-Replication (i.e., $1 < k < K$) can be the 'best':

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/10.1145/2896377.2901499

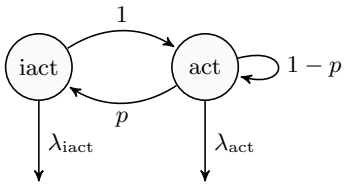


Figure 1: Two-state Markov chain $Z(n)$

No-Replication: Uniform: Assuming uniformly distributed service times, i.e., $x_i \sim \mathcal{U}_{[0,1]}$, replication is detrimental:

$$k\mathbb{E}\left[\min\left\{x_1,\ldots,x_k\right\}\right] = \int_0^1 kx^k dx = \frac{k}{k+1} \ .$$

Full-Replication: Weibull: Assume Weibull distributed service times, i.e., $f(x) = e^{-(x/\lambda)^\alpha}$. For $\alpha < 1$, a higher degree of replication is 'better':

$$k\mathbb{E}\left[\min\left\{x_1,\ldots,x_k\right\}\right] = k\frac{\lambda}{k^{1/\alpha}}\Gamma(1+1/\alpha) \ .$$

Partial Replication: Pareto: Assume Pareto distributed service times, i.e., $f(x) = x^{-\alpha}$ for $x \geq 1$. For suffiently small $\alpha > 1$, Partial-Replication is 'best':

$$k\mathbb{E}\left[\min\left\{x_1,\ldots,x_k\right\}\right] = k + \frac{k}{k\alpha - 1} \ .$$

This last example highlights that the performance of replication strategies heavily depends on the replication factor k, the service time distribution, and other underlying assumptions. In particular, performance is not monotonic in k, and thus an optimization framework is desirable.

3. THEORY

For simplicity, we assume that K is an integral multiple of k. Further, the jobs are assigned to the $\frac{K}{k}$ batches in a round robin scheme, i.e. the interarrival times for one batch can be described as:

$$\tilde{t}_i := \sum_{j=0}^{K/k-1} t_{(i-1)\frac{K}{k}+j} \ .$$

The *steady-state* response time r has a representation:

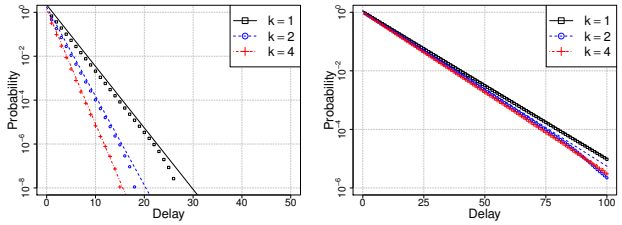
$$r =_{\mathcal{D}} \max_{n\geq 1}\left\{\sum_{i=1}^{n+1}\min_{j\leq k}\{x_{i,j}\} - \sum_{i=1}^{n}\tilde{t}_i\right\} \ . \qquad (3)$$

We consider the following correlation model for the replicas (from [4]):

$$x_{i,j} = \delta y_i + (1-\delta)\, y_{i,j} \ ,$$

where the random variables y_i and $y_{i,j}$ are i.i.d. Here, the parameter $\delta \in [0,1]$ describes the degree of correlation amongst the replicas; $\delta = 0$ corresponds to the i.i.d. case.

For the interarrival times, we consider two scenarios. In the first scenario the $\{t_i\}$ are i.i.d., in the second scenario they have a Markovian structure as in Figure 1: the Markov chain $Z(n)$ alternates between *active* and *inactive* periods; while active, exponentially distributed interarrival times are generated (parameter λ_{act}). While inactive, *one* interarrival time (exponentially distributed, parameter $\lambda_{\text{inact}} < \lambda_{\text{act}}$) is generated, and the chain jumps back to the active state.



(a) i.i.d. scenario (b) Markovian scenario

Figure 2: Stochastic bounds vs. simulation results accounting for 10^9 packets ($K = 4$, $\rho = .75$, $\mu = 1$, $\delta = .5$)

For $0 \leq \theta < \lambda_{\text{iact}}$, let T_θ denote the following matrix:

$$T_\theta := \begin{pmatrix} 0 & \frac{\lambda_{\text{act}}}{\lambda_{\text{act}}+\theta} \\ p\frac{\lambda_{\text{iact}}}{\lambda_{\text{iact}}+\theta} & (1-p)\frac{\lambda_{\text{act}}}{\lambda_{\text{act}}+\theta} \end{pmatrix} \ .$$

Further, let $\xi(\theta)$ denote the spectral radius of T_θ, and $h = (h_{\text{act}}, h_{\text{iact}})$ be a corresponding eigenvector.

THEOREM 1. *For the two scenarios above, let*

$$\theta_{iid} := \sup\left\{\theta \geq 0 \;\middle|\; \mathbb{E}\left[e^{\theta\delta y_i}\right]\mathbb{E}\left[e^{\theta(1-\delta)\min_{j\leq k}\{y_{i,j}\}}\right] \mathbb{E}\left[e^{-\theta t_i}\right]^{\frac{K}{k}} \leq 1\right\}$$

$$\theta_{mkv} := \sup\left\{\theta \geq 0 \;\middle|\; \mathbb{E}\left[e^{\theta\delta y_i}\right]\mathbb{E}\left[e^{\theta(1-\delta)\min_{j\leq k}\{y_{i,j}\}}\right] \xi^{\frac{K}{l}}(\theta) \leq 1\right\}$$

Then for the response time holds for all $\sigma \geq 0$:

$$\mathbb{P}(r_{iid} \geq \sigma) \leq \mathbb{E}\left[e^{\theta_{iid}\min_{j\leq k}\{x_{1,j}\}}\right] e^{-\theta_{iid}\sigma}$$

$$\mathbb{P}(r_{mkv} \geq \sigma) \leq \mathbb{E}\left[e^{\delta\theta_{mkv}y_i}\right]\mathbb{E}\left[e^{(1-\delta)\theta_{mkv}\min_{j\leq k}\{y_{i,j}\}}\right] e^{-\theta_{mkv}\sigma}$$

To numerically compare our stochastic bounds to simulation results we refer to Figure 2; we remark that in both considered scenarios, the bounds are remarkably accurate.

This work was partially funded by the DFG grant Ci 195/1-1.

4. REFERENCES

[1] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, Jan. 2008.

[2] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, and E. Hyytiä. Reducing latency via redundant requests: Exact analysis. In *ACM Sigmetrics*, pages 347–360, 2015.

[3] G. D. Ghare and S. T. Leutenegger. Improving speedup and response times by replicating parallel programs on a snow. In *10th International Conference on Job Scheduling Strategies for Parallel Processing (JSSPP)*, pages 264–287, 2004.

[4] G. Joshi, Y. Liu, and E. Soljanin. On the delay-storage trade-off in content download from coded distributed storage systems. *IEEE Journal on Selected Areas in Communications (JSAC)*, 32(5):989–997, May 2014.