Data Replication for Reducing Computing Time in Distributed Systems with Straggles

Amir Behrouzi-Far and Emina Soljanin
Department of Electrical and Computer Engineering, Rutgers University
Piscataway, New Jersey 08904, USA
{amir.behrouzifar,emina.soljanin}@rutgers.edu

Distributed computing has been given a great attention for several reasons, including big data processing [1]. However, implementing computing algorithms in distributed systems introduces new challenges that have to be addressed in order to benefit from the paralleziation. In particular, since the failure rate and/or slow down of a system increase with the number of computing nodes, robustness is an essential part of any reliable distributed computing/storage algorithm [2]. For achieving robustness, redundancy is proposed in the literature [3]. Redundancy enables a task master to generate the overall result form the computations of only a subset of all computing nodes, instead of all of them. Thus, slow workers, known as stragglers, can be mitigated [4].

In spite of a growing body of work on redundancy in distributed systems, we are still far from fully understanding its benefits and costs. In particular stragglers, that are the consequence of resource contention, network congestion, Input/Output (I/O) operations, etc., could be exacerbated by a non-careful redundancy planning. Although understanding the exact benefits and costs requires detailed knowledge about both the system and the computing job, many studies have been devoted to performance evaluation of these systems under reasonable modeling.

In this work, we consider a distributed system with master-worker architecture. A computing job is an executable file which has to operate over a possibly large data set. We assume that the executable can be concurrently run on different workers, each of which hosting a subset of the original data set. This model is well applicable to a wide range of problems, e.g model training in supervised learning. We study the completion time in the aforementioned system, defined by the total waiting time for the master to generate the overall results from a subset of local compute results.

We study two general methods for data replication across workers: overlapping batches and non-overlapping batches. With overlapping batches, the data subset at each worker either completely overlaps or does not overlap at all with the subsets at other users. With overlapping batches, on the other hand, the data set at any worker has partial overlaps with subsets at other workers. Our first result is that, a balanced distribution of non-overlapping batches results in minimum expected computing time, if the service time distribution of workers is stochastically decreasing and convex random variable. We then study the effect of the degree of redundancy on the completion time, for several service time distributions of workers. The redundancy level could vary from no-redundancy (full parallelism) to full-redundancy (full diversity).

In particular, we studied the behavior of the average computing completion time for three different widely studied service time distributions of workers: Exponential, Shifted-Exponential and Pareto. We analytically showed the following:

- With the ***Exponential*** service, full diversity minimizes the expected completion time and its variance.
- With the ***Shifted-Exponential*** service, the optimum degree of redundancy depends on the distribution's parameter, as follows. If workers' service time follow $SExp(\Delta, \lambda)$, the larger the product $\Delta\lambda$ the higher parallelism is optimal.
- With the ***Pareto*** service time, the optimum degree of redundancy depends on the distribution's parameter, as follows. If workers' service time follow $Pareto(\sigma, \alpha)$, the larger the α the higher parallelism is optimal.

In addition to modeling and analysis, we measured the effect of redundancy on the performance of Google's compute jobs, for which we collected data from Google cluster traces. First of all, we observed that the service time of the jobs in Google show both heavy-tail, like Pareto, and exponential-tail, like Shifted-Exponential behavior. We showed that Google compute jobs could benefit significant performance gain once they run with redundancy. Nevertheless, the degree of redundancy is an important decision metric, as large redundancy can deteriorate the performance, which shows the importance of our analysis on optimal degree of redundancy.

References

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," 2004.
[2] J. Dean and L. A. Barroso, "The tail at scale," *Communications of the ACM*, vol. 56, no. 2, pp. 74–80, 2013.
[3] Y. Brun, G. Edwards, J. Y. Bang, and N. Medvidovic, "Smart redundancy for distributed computation," in *2011 31st International Conference on Distributed Computing Systems*. IEEE, 2011, pp. 665–676.
[4] A. Behrouzi-Far and E. Soljanin, "On the effect of task-to-worker assignment in distributed computing systems with stragglers," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2018, pp. 560–566.