Hopper: Decentralized Speculation-aware Cluster Scheduling at Scale – Public Review

Lixin Gao
University of Massachusetts
Amherst, MA
lgao@ecs.umass.edu

The huge volume of data available today has led to interest in parallel processing on commodity clusters. Data analytics distributed frameworks such as Hadoop, Spark, or Pregel are designed for parallel processing of a large amount of data. These frameworks break a computation job into small tasks that run in parallel on multiple machines, and aim to scale to very large clusters of inexpensive commodity computers. However, as jobs increase in size and complexity, scheduling these jobs so as to provide scalable and predictable performance becomes challenging.

While job scheduling is a classic problem that has been studied extensively in the context of parallel processing, scheduling data analytics jobs in large clusters of inexpensive commodity computers are complex and hard to model. In particular, it is possible for a compute node to perform poorly. As a result, tasks running on the node make progress slowly and become ”stragglers”, leading to significant delay in job completion time. Speculative execution, *i.e.*, running a speculative copy of a task on a different node, has been deployed in many data analytics frameworks to mitigate the ”straggler” effect.

This paper is the first to propose to schedule original and speculative tasks of a job together, instead of considering original tasks only. It identifies that speculation is a commonplace in production (25% of all tasks). The key idea of the proposed scheduler, Hopper, is that tasks are given more slots to execute speculative tasks, if the cluster resource is sufficient. Exact amount of slots allocated to a task depends on the size of the job as well as how busy the cluster is. Further, a decentralized scheduling that approximates the global state of the cluster through querying two workers (i.e., the ”power of two choices” proposed for decentralized load balancing) is described and shown to nearly match the performance of the centralized implementation. With such a speculation-aware schedule, the overall performance of the centralized and decentralized implementation is improved significantly (50-60%).

All reviewers like the main idea of Hopper that takes speculation into account in scheduling decisions. The reviewers are in particular excited about the observation that the gain provided by extra computation slots exhibit a sharp knee, which is an effective threshold for determine the amount of slots allocated to each task. The resulting scheduling policy is conceptually elegant and is backed by theoretical analysis (albeit with simple models).

Hopper’s impressive performance hinges on the assumption that the job size follows the heavy-tailed Pareto distribution. While this assumption holds true in production traces such as Facebook and Bing traces, it is not clear this is true in the future. It would be interesting to investigate various job size distributions and understand whether they exhibit the knee as in the case of the heavy-tailed Pareto distribution. It would also be helpful to provide the methodology to determine the knee if it exists, given the central role that it plays in the design of the scheduling policy.

In summary, Hopper takes the first step in considering speculative and original tasks together in scheduling of cluster jobs. While the proposed scheduler perform very well for heavy-tailed job size distribution, the paper raises interesting questions about when and how speculation-aware scheduling can be effective.