Pandas: An Efficient Priority Algorithm for Near-Data Scheduling

Qiaomin Xie
ECE, UIUC
In collaboration with Yi Lu, Mayank Pundir, Cristina Abad

Abstract

The prevalence of data-parallel applications has made near-data scheduling an important problem. An example is the map task scheduling in the map-reduce framework. The challenge lies in that the processing rate depends on the task-node pair, which constitutes a hard affinity scheduling problem. Moreover, despite the uniform placement of data, the data being processed at a given time often concentrates on a subset of nodes, causing hot-spots and prolonged exacerbation of job completion times. The existing Hadoop schedulers do not solve the affinity-scheduling problem, nor handle hot-spots.

We propose Pandas (Priority Algorithm for Near-DAta Scheduling) and integrate it into the Hadoop FIFO and Fair Schedulers. Pandas serves to balance data locality and cluster utilization so that the system throughput is drastically improved over all load conditions. In particular, high data locality is achieved when no hot-spots are present and high utilization is achieved when hot-spots occur. The main idea is load balancing on local tasks, which at the same time serves as an estimate to assist the decision on remote task assignment. Together, the overall system throughput is drastically increased.

On the theoretical side, assuming an exponential distribution for task processing times, we show that Pandas is throughput-optimal [1], i.e., it is robust to variation in job inter-arrival times, and can accommodate any load if there exists an algorithm that can accommodate it without making the system unstable. We have also shown that Pandas is heavy-traffic optimal in all traffic scenarios, i.e., it asymptotically minimizes the average delay as the system becomes critically loaded. This makes Pandas the only known heavy-traffic optimal algorithm for this problem.

For loads away from the heavy-traffic region, we evaluate Pandas in a variety of environments including a private cluster, Amazon's Elastic Compute Cloud (EC2) and via large-scale simulations. We use both long traces with mixed loads and traffic distributions, as well as short traces with fixed loads to investigate performance details [2]. Evaluation with production traces shows that the Pandas-enhanced FIFO Scheduler achieves up to 11-fold improvement over the FIFO Scheduler, and the Pandas-enhanced Fair Scheduler achieves up to 22-fold improvement over the Fair Scheduler in terms of the average job completion time. In addition, the improvement in completion time is experienced by jobs of all sizes. With both FIFO and HFS, deferring job priority to task priority only slows down a very small fraction of jobs, with almost all jobs experiencing a significant improvement in completion times.

References

[1] Q. Xie, and Y. Lu. Prioprity Algorithm for Near-Data Scheduling: Throughput and Heavy-Traffic Optimality. To appear in *IEEE INFOCOM*, 2015.

[2] M. Pundir, Q. Xie, Y. Lu, C. L. Abad, and R. H. Campbell. Pandas: An Efficient Priority Algorithm for Near-Data Scheduling. *Submitted*, 2014.