LOG8415
Advanced Concepts of Cloud Computing

Foutse Khomh
S. Amirhossein Abtahizadeh
Département Génie Informatique et Génie Logiciel
École Polytechnique de Montréal, Québec, Canada
`foutse.khomh[at]polymtl.ca`
`a.abtahizadeh[at]polymtl.ca`

1 Identification

Student's name: Chun-An Bau

Date of the reading note: Oct 6, 2021

Author(s):   Andy Goder, Alexey Spiridnov, Yin Wang

Title of the article: Bistro: Scheduling Data-Parallel Jobs Against Live Production Systems

Publication: 2015 USENIX Annual Technical Conference (USENIXATC 15). 2015

2 Article

Keywords: Big Data Processing, Parallel Computing, Job Scheduler

Concepts and definitions:

- Data shard: A piece of data segment

- Task: the work to be completed on one data shard

- Job: the overall work to be completed on a set of data shards

- BLOB: Binary Large OBject, which is regarded as a collection of binary data stored as a single entity

- Data host: The computer that stores data shards.

- Map only/heavy job: A job where each task operates on its own data shard independently or relies heavily on the mapper. These jobs can mostly be separated into a number of parallel tasks without problem

- Worker: The process working on tasks

- Scheduler: The process assigning tasks

- Data-intensive: Task or job that required time or resource regarding I/O instead of computing in processors

- Batch job: A predefined group of processing actions submitted to the system to be performed with little or no interaction between the user and the system

Summary:

Since Facebook needs to store lots of data with distinct formats, how to store, process, and transfer data has bothered the engineers for years. Although one of the large-scale processing approaches, Hadoop, provides robust functionalities and works well with its own storing system, HDFS, there are some drawbacks cannot be ignored, such as:

- Using inefficient way to store offline data, which only a small portion will be used or even none of them will be used.

- Data consistency problem, such as bulk database updates must mutate the online data.

Moreover, many other approaches have the limitations like assuming data is immutable whereas online data changes frequently, not supporting hard constraints on the burdens that jobs place upon data hosts, or assuming a specific offline data ecosystem where the systems actually access a wide range of data sources and storage systems. Thus, the engineers developed a novel scheduler system, Bistro, with outstanding features like:

- Allows offline batch jobs to share clusters with online customer-facing workloads without harming the performance of either.

- Reduces infrastructure hardware by eliminating the need for separate **online** and **batch** clusters while improving efficiency by eliminating the need to copy data between the two

- Easy to scale both jobs and clusters relative to one another, and allows concurrent scheduling for better throughput

Bistro has replaced Hadoop as the primary scheduler system for years, which has processed trillions of rows and petabytes of data.

Research contributions:

- Define a class of data-parallel jobs with hierarchical resource constraints at online data resources

- Describe Bistro, a novel tree-based scheduler that safely runs such batch jobs "in the background" on live customer-facing production systems without harming the "foreground" workloads

- Compare Bistro with a brute-force scheduling solution, and describe several production applications of Bistro at Facebook

- Bistro is available as opensource software

3 Analysis

Quality:

General organization:	Language and style:	Technique:	Bibliography:
☐ Very good;	☐ Very good;	■ Very good;	■ Very good;
■ Good;	■ Good;	☐ Good;	☐ Good;
☐ Medium;	☐ Medium;	☐ Medium;	☐ Medium;
☐ Bad;	☐ Bad;	☐ Bad;	☐ Bad;
☐ Very bad.	☐ Very bad.	☐ Very bad;	☐ Very bad;
		☐ N/A.	

Forces of the message:

- It is definitely a robust system that may inspire companies with similar requirements, like large-scale map only/heavy tasks.

- The research proves its value by being used by one of the biggest software company, Facebook.

Weaknesses of the message:

- If the Implementation chapter is discussed in the same order of how the system really works, it would be better for the reader to understand.

- The comparison to Hadoop, another robust related work, is too less (two graphs). If more comparison was added, the research could be more convincing.

Future directions:

- How can the load balancer help to improve the performance of the system?

- Does the system also suitable for other scenarios or companies?

Other important articles:

- Job-aware scheduling in eagle: Divide and stick to your probes.
  Proceedings of the Seventh ACM Symposium on Cloud Computing, 2016, pp. 497-509