Parallel Batch Processing



VS





Experiment

Motivation:

Evaluating the throughput and runtime performance of spark and flink for different data sizes.

We are expecting spark to perform better with higher Throughput and lower runtime compared to flink.

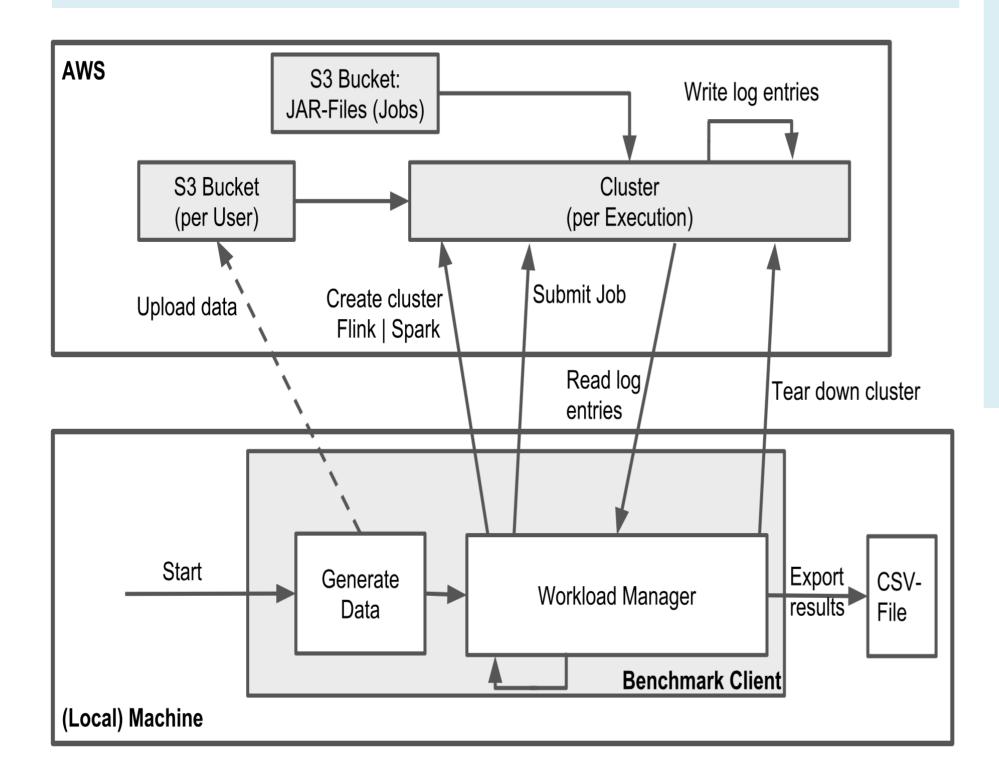
Research Question:

Measure throughput and runtime (for Group-By, Sorting, Aggregate) of Flink and Spark clusters on different input sizes and fixed structures (Book-Store) on fixed comparable systems and comparing the results.

Expectation

Expectation:

We are expecting Spark to perform better with higher throughput and lower runtime compared to Flink [1].



Interpretation of the Results

The Spark throughput for each job (GroupBy, Aggregation, Sorting) and for each data size (1GB, 5GB, 10GB) was higher. The Spark runtime for each job (GroupBy, Aggregation, Sorting) and for each data size (1GB, 5GB, 10GB) was faster.

Conclusion

As expected, the Spark performance for the throughput and runtime was always better.

We conducted the experiments with small data sizes compared to real-world problems (>500GB). This was due to the AWS credit limitation (100\$). Further experiments with bigger data sizes are needed to check if the Spark performance is also better in these cases.

[1] https://www.dbgroup.unimore.it/tesi/Magistrale/201516 Andrea Spina presentazione.pdf

Workloads & Metrics

Data:

- typical book store
- Amount of data size -> 1GB, 5GB, 10GB
- Amount of records: 9m, 44m, 88m

Workloads for each metric and SUT:

- a) GroupBy: genre, count per group
- b) Sorting: number of pages (ascending)
- c) Aggregation: price (max)

Metrics:

- 1. Runtime
- 2. Throughput

System Under Test (SUTs)

Benchmark approach:

client perspective

SUTs:

- Flink 1.6.2
- Spark 2.4.0

Cluster settings:

- Provider: AWS
- One master; Two slaves
- EC2 type: m5.xlarge (4 vCore, 16 GiB memory)
- us-east-1 region (N. Virginia)

