CS 744 Assignment 1 Report and Learning

Group 6: Zichuan Tian, Xinyi Li, Jing-Yao Chen

Part1:

We deployed HDFS to Ubuntu 16.04 ext4 and configured Spark to run on top of it. We observed that there is a HDFS master node and HDFS secondary master node running on one of the VM, and another two worker nodes running on two VMs.

Part2:

We wrote a small Spark program to sort a CSV file based on column keys. We didn't observe any difficulty running the code locally, but when we try to start HDFS and load file to HDFS then run on it, we observed that the output data were partitioned and difficult to recombined. We used the coalesce function to combine all output partitions.

Part3:

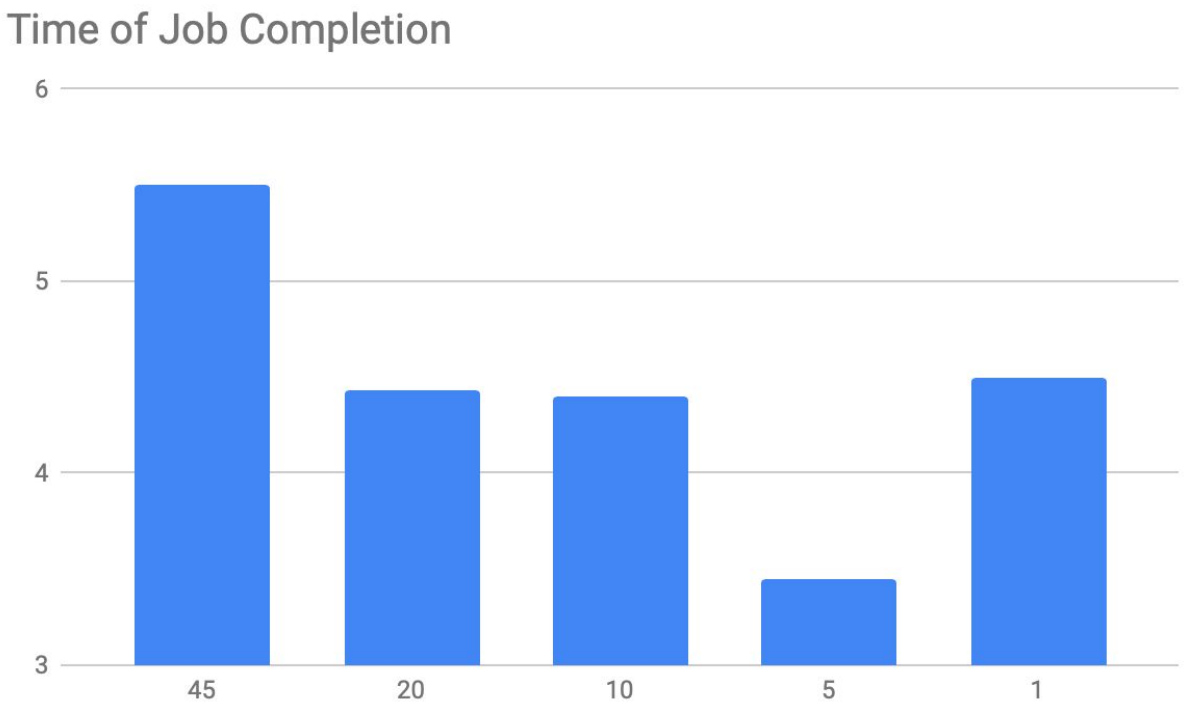
Task 1: PageRank Implementation

To reduce the number of keys, all strings are converted to lowercase. Dirty data is filtered out prior to ranking. The algorithm to do pagerank is below:

Task 2: Effect of Custom RDD Partitioning

Iteration = 10 with RDD persist at MEMORY level

PartitionBy()	Application Complete Time
55 (Default)	4.49
1	4.50
5	3.45
10	4.40
20	4.43



We observed that a partition size of 5 is optimal for page ranking this data set. Of course the number of partitions depends on a lot of factors, such as number of core/physical threads on the machine, memory I/O bandwidth and latency, type of workflow and etc.

Task 3: Effect RDD Persistence

Without RDD Persist:

- Application complete time: 5 mins 22 seconds.

With RDD Persist:

- Application complete time: 4 mins 40 seconds.

With RDD persist, the memory is copied to a cache located on each worker. As a result, each action does not need to redo all operations from the beginning so the completion time is reduced.

Task 4: Fault-Tolerance By Killing Worker Process

Without worker fail:

- Application complete time: 4 mins 40 seconds.

With work fail:

- Application complete time: 4 mins 48 seconds.

With a worker process terminated, it is natural that the completion time is longer compared to systems without fault. However, we did not observe a significant difference for this.

Appendix:

There is one run.sh for each task, in each run.sh:

```
spark-submit page_rank.py $1 $2
```

The first argument is input file path (hdfs://IP:PORT/path_to_input) and the second argument is output file path.

To run

```
./run.sh input_file_path output_file_path
```