CprE 419 Lab 4: Sorting Using Hadoop MapReduce

Shuo Wang

My algorithm is using two M-R rounds:

1. The first M-R round samples the input data by a fixed ratio and output the 1st quartile, the median and the 3rd quartile of the sampled keys as the partitioning break points. Those calculated partitioning break points would roughly split the input file into four parts evenly.

2. The second M-R round use the calculated partitioning break points from the first round to customize the partitioner. Every key in the smallest bracket will be labeled as 0 by the partitioner and sent to the first reducer; every key in the second smallest bracket will be ladeled as 1 and sent to the second reducer, so on and so forth. Then each reducer get the inputs and write them out in a sorted manner and the final 4 files will be in a total order.

The first round uses 1 reducer and the second round uses 4 reducers.

The two rounds are written in separate Java files and run separately. Once the first round gets the partitioning points, hard code the partitioner for the second round and run it to get the final sorted files. For different input files, it requires to manually check the output from the first round and code the partitioner in the second round.

Customizing the partioner based on the input file automatically rather than hard coding was also tried.

Two ideas have been tested:

1. Write the output from the first round into HDFS and read it out in the second round. Since the partitioner can recognize nothing but the key-value pairs from mapper, the only way to pass the customized parameter to partitioner is read the HDFS file and construct the parameter inside the partioner class. By doing this, it turns out that each time a key-value pair gets labeled the reading file function is called once. As consequence if there input file is reasonably large enough there will be too many open files and the job will crash.

2. Use hadoop built-in TotalOrderPartitioner. There are two potential ways to set up the partition file for the TotalOrderPartioner:

    a. from InputSampler. Since the shuffling and partitioning process need to begin right after the mapping process, the partitioning strategy needs to be prepared before the mapping ends. So than the InputSampler samples the input keys of mapper and apply the found strategy to the output keys of mapper. This mechanism is not good since most of the time the input keys of mapper differ from the output keys of mapper. So most of the time, this method is not preferred. During my work, I tried KeyValueTextInputFormat to meet the requirement discussed above but some runtime error about "type mismatch" showed up.

    b. create the partition file by ourselves. I tried to use one extra M-R round to write the sampled keys into a file but runtime error showed that the self-created file is

not readable for the TotalOrderPartitioner.

The separate output log files of my program running on the largest file (250M records) were included in the submission.

The result of my program on the largest file (250M records):
Round 1: 41 s

```
1424801340152_0149
15/02/25 23:00:49 INFO mapreduce.Job: Job job_1424801340152_0149
running in uber mode : false
15/02/25 23:00:49 INFO mapreduce.Job:  map 0% reduce 0%
```

….

```
15/02/25 23:01:25 INFO mapreduce.Job:  map 100% reduce 67%
15/02/25 23:01:28 INFO mapreduce.Job:  map 100% reduce 100%
15/02/25 23:01:30 INFO mapreduce.Job: Job job_1424801340152_0149
completed successfully
```

Round 2: 26 min 51 s

```
1424801340152_0150
15/02/25 23:13:40 INFO mapreduce.Job: Job job_1424801340152_0150
running in uber mode : false
15/02/25 23:13:40 INFO mapreduce.Job:  map 0% reduce 0%
15/02/25 23:13:54 INFO mapreduce.Job:  map 1% reduce 0%
```

….

```
15/02/25 23:40:03 INFO mapreduce.Job:  map 100% reduce 100%
15/02/25 23:40:30 INFO mapreduce.Job: Job job_1424801340152_0150
completed successfully
15/02/25 23:40:31 INFO mapreduce.Job: Counters: 51
```

Total time: 27 min 32 s

From the size of the four output files, the load of each reducer has been balanced well.

```
[shuowang@n0 ~]$ hls /scr/shuowang/lab4/exp1/output
Found 5 items
-rw-rw-r--   3 shuowang 419x          0 2015-02-25 23:40 /scr/shuowang/lab4/exp1/output/_SUCCESS
-rw-rw-r--   3 shuowang 419x 6384520197 2015-02-25 23:40 /scr/shuowang/lab4/exp1/output/part-r-00000
-rw-rw-r--   3 shuowang 419x 6158511117 2015-02-25 23:28 /scr/shuowang/lab4/exp1/output/part-r-00001
-rw-rw-r--   3 shuowang 419x 6087509010 2015-02-25 23:32 /scr/shuowang/lab4/exp1/output/part-r-00002
-rw-rw-r--   3 shuowang 419x 6119459676 2015-02-25 23:35 /scr/shuowang/lab4/exp1/output/part-r-00003
[shuowang@n0 ~]$
```