CPrE/SE 419: SOFTWARE TOOLS FOR LARGE-SCALE DATA ANALYSIS, SPRING 2024

LAB #3: SORTING

Purpose

In this lab, your goal is to write a program using MapReduce that can sort a large data set quickly. Sorting is a common task on large data sets, and you will be working on a standard input data set, which is used as a benchmark for comparing different implementations of sort.

At the end of the lab, you will be able to write an algorithm and implementation for sorting a large data set and measure its performance on various input data sets.

Constraints:

- Use no more than 10 reducers at any MapReduce stage.
- In your program, make sure to delete all the temporary directories which contain the intermediate data before exiting.

Metric of Success:

- The output of your program could be one or more sorted lists. Suppose you have *n* reducers, e.g. R1, R2, …, Rn, all data in R1 need to be smaller than all data in R2, and all data in R2 need to be smaller than R3 and so on.

Submission (50 points)

Create a zip (or tar) archive and hand it in through Canvas. Your submission includes:

- A writeup report:
  - The results for each experiment, including the top 5 and last 5 lines for each partition.
  - The strategy you used for experiment 2.
  - The arguments for your jar file to run, for each experiment.
  - Any other resources to support your work.
- Commented Code for your program. Include all source files needed for compilation.

Resource

- Lecture notes on sorting and partitioners in MapReduce
- http://hadoop.apache.org/docs/r3.1.1/api/org/apache/hadoop/mapreduce/Partitioner.html
- https://hadoop.apache.org/docs/r3.1.1/api/org/apache/hadoop/mapreduce/lib/partition/TotalOrderPartitioner.html

Template Code

- sortingTemplate.java on Canvas (Exp 1)
- customSort.java on Canvas (Exp 2)
- Chapter 9 MapReduce Features: Total Sort, Hadoop-The Definitive Guide

Dataset

We have multiple datasets in cybox, https://iastate.box.com/s/3l34r2d3yvssxzzbzxvfn1zwt9yve0uk , to play with, such as "input-50k", "input-500k" and etc. Each dataset is made of multiple lines, one line per record. Each record has a key (first 15 chars) and a payload (40 chars), separated by tab character. The final output should be **sorted by key**, and the output should also include the payload for the key. Example of the input data:

```
NVKqt8PQj3mBOuy MK8qIjdRmH2brzSdgsVVV1vBySDKV4Bi2WAVQD8S
G1LiDI2qppbSejO EfdsyylTGt3HxHElvbbQbacOCEL2UgdALZtHsHuH
qsTfbll2pdQ0vJa X43hsXk2ZL1ZSB5jtfE60oDYoXYBvi1WCIXXjGj2
c1PMEsF9RcrUZpk BkUJDW7KQtKBJJs8kg7jrQdDNVBW09uV5ebbMolH
9by7zR2tVXQmkFb MyOVQH6pln6Z5IRVem0PHJPZOZ7nAOgzXJncVmhk
7QX87mR4VJK31nX Nxv5PeUxpSesur6dRD0oC0OjxoZY2Yg9Zr3rNB6Z
Z9oXD8C9hPuurV4 KRAooUxQGyAT0toze5Dbug4oPb0gSO3qE53Yva0y
3C1f3RzmhUNyea7 cCgS14iBV4AKYBcthjoeVP4e8t0G1RIuBPlYQdzI
1A7KJRRshpqr7fd 2VovQIUHtXfGdiO84DnkDBMmO9ZSttVgxjcHCihi
AE3yMsPvmjGgHeV r04diBqll1J5AJx9qPlN26t9ueNogJ7MNtp8zOKp
5x567ZMSciNN5me aLmCX6RfUFpv02jmap1B8foZlJVHJPZfxKVcukKf
qMAH3YeuKvWVcvn Lz8FQl6tWpF41iTYAHsBCsgxSD0YuVMkKpREALhC
```

Experiment 1 (15 points)

Sort the data in "input-5m" by keys, using TotalOrderPartitioner and InputSampler. Use up to 10 reducers.

Experiment 2 (35 points)

Write your own "MyPartitioner" class and use it to sort the data in input-500k. You MUST NOT use the TotalOrderPartitioner or the InputSampler class. Explain your algorithm/strategy for partitioning and sampling in the submission. Use up to 10 reducers.

Bonus (5 points)

Question: We have a list of datasets ("input-5k", "input-50k", "input-500k", "input-5m"), sorted by their size. Try TotalOrderPartitioner and your own partitioner (MyPartitioner) on each dataset. Do you get different results? What makes it different?

Which is the largest dataset your solution can sort? You can get the bonus if you can sort 5m dataset by using only up to 10 reducers and your own partitioner.