CprE 419 Lab 4: Sorting Using Hadoop MapReduce

Department of Electrical and Computer Engineering
Iowa State University
Spring 2015

Purpose

This is the third lab on MapReduce. 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.

Goal

Write a MR program to sort a large amount of data quickly.

Constraints:

- Use less than 4 reducers at any MapReduce stage
- Your code must be runnable for any user. This means you cannot hardcode your path references, say for the output directory or temporary directories in your code. They must be passed in via the command line.

Metric of Success:

- The output of your program could be one or more sorted lists. Supposed you have r reducers, e.g. f1, f2, …, fr, the data in f1 should be the smallest; the data in f2 should be smaller than f3, …, fr, and so on.
- For grading purposes we will be testing your code so again, it must be runnable for any user.
- The performance of algorithm should be measured by taking the sum of the runtimes of the **total MapReduce procedure**. If you have several stage of MapReduce, these runtime should be counted in your runtime.

Your goal is to minimize the total reducer time. The student with the fastest sorting algorithm on our largest input dataset will receive **20 points of extra credit**.

Dataset

We have uploaded multiple datasets for testing purposes in the directory "/class/s15419x/lab4". We want you to submit the performance of your program on the largest dataset in this directory. The name of the data file ends with the number of records in the input, for example "gensort-out-50M" has 50 million input records. Since each record is 100 bytes, this file takes about 5GB.

Each dataset is made up of multiple lines, one line per record. Each record has a key of 10 characters, followed by a space, followed by a payload (which in turn has a sequence number plus some other data). The final output should be sorted by key, and the output should also include the payload for the key.

Submission

Create a zip (or tar) archive with the following and hand it in through blackboard**.** Label your submission “lab4_<yourNetID>”

- A writeup describing your algorithm and anything interesting.
- A screenshot of the total reducer time for each MR round. You can put this in the lab write up
- Commented Code for your program. Include all source files needed for compilation.
- Executable JAR.
- A readme.txt that describes what the arguments are for your jar file to run.

Resource

- https://hadoop.apache.org/docs/r2.4.1/api/org/apache/hadoop/mapred/Partitioner.html