PolyU COMP4434 Assignment 1

Introduction

Goal

By doing this assignment, you will understand how to use Hadoop MapReduce

Background

Write a MapReduce program to compute the product C of two matrices A and B.

Basic idea

$Suppose\ C = A \times B$

$A\ has\ L\ rows\ and\ M\ columns$

$B\ has\ M\ rows\ and\ N\ columns$

In the map phase:

    a. For each element (i, k) of A, emit $\left((i, j), A_{ik}\right) for\ j\ in\ 0..N - 1$

    b. For each element (k, j) of B, emit $\left((i, j), B_{kj}\right) for\ i\ in\ 0..L - 1$

In the reduce phase:

    a. Key = (i, j)

    b. Value = $\sum_k A_{ik} * B_{kj}$

Then we get the final result.

Input file format

We use only one plain text file to represent two matrices, each line follows these rules:

- the content is A, i, j, v or B, i, j, v , **no spaces between** them
  - A indicates the left matrix of a multiplication
  - B indicates the right matrix
- A, i, j, v means $A_{ij} = v$
- $i\ and\ j\ starts\ with\ 0$
- $If\ v\ is\ 0$, the text file doesn't have that line -- it is useful to represent a sparse matrix

Suppose we want to represent this multiplication

$$\begin{bmatrix} 1 & 3 & 4 & -2 \\ 6 & 2 & -3 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & -2 \\ 4 & 3 \\ -3 & -2 \\ 0 & 4 \end{bmatrix}$$

Our input file will be this:

A,0,0,1
A,0,1,3
A,0,2,4
A,0,3,-2
A,1,0,6

A,1,1,2
A,1,2,-3
A,1,3,1
B,0,0,1
B,0,1,-2
B,1,0,4
B,1,1,3
B,2,0,-3
B,2,1,-2
B,3,1,4

Notice that the file doesn't contain the line B,3,0,0

Output file format
The computed result should be like this:
0,0,1
0,1,-9
1,0,23
1,1,4

It represents this result matrix

$$\begin{bmatrix} 1 & -9 \\ 23 & 4 \end{bmatrix}$$

How to run the program
It is very similar to the word count program, you need to export to a jar file first, say matrix.jar, then run it with some parameters. You need 6 parameters for this program, like this:

~/Programs/hadoop-1.2.1/bin/hadoop jar ~/guestshare/matrix.jar 2 4 4 2
/user/bigdata/matrix/input /user/bigdata/matrix/output

- the first parameter (**red color**): means the left matrix has **2** rows
- the second parameter (**green color**): means the left matrix has **4** columns
- the third parameter (**orange color**): means the right matrix has **4** rows
- the fourth parameter (**black color**): means the right matrix has **2** columns
- the fifth parameter (gray color): means the input folder on HDFS
- the sixth parameter (**purple color**): means the result folder on HDFS

To begin with

You are given three sets of files:
- ReadMe.pdf (this file)
- **data** folder, it contains two groups of matrices multiplication
  - matrix1.txt and result1.txt are the first group

- matrix1.txt is the input which describes the two matrices
- result1.txt is the result for you to check later
  - matrix2.txt and result2.txt are another group
  - **You can use these two groups to check your program, but we will use other test files to grade your program.**
- **src** folder, a file MatrixMultiplication.java has been prepared for you --- we have already done some tedious job for you such as parameters reading and passing. **Your job is to implement correct map and reduce function, <u>don't modify any given parts of the file or otherwise your program will not work correctly</u>.**

What to submit

Make a zip file called matrix.zip which contains only MatrixMultiplication.java file, submit it through blackboard.

Grading policy

We will export a jar file from your source file like we did for word count, then execute it with 6 parameters, then check whether the result is correct.
We will use several different input files to test your program, say we use 10 different matrices multiplication input files, if your program only passed 8 of the 10, you get 80.

Deadline: <mark>9 Feb 2015, 14:59</mark>

Late Penalty

late x day: your score = raw score * (100 – 20 * x)%

Plagiarism

It is easy to detect the similarity of source files, and cases will be strictly handled according to the University's regulation, so please don't risk doing that.

Questions?

Ask your teaching assistant during the lab, ask Eric during lecture breaks.