Project 1

Instructions

Part 1. [6 points] Set up a single node cluster and optionally an eclipse development environment to create and test your programs.

(a) Get VMWare, VirtualBox and so on (install)

(b) Get Cloudera (install)

(c) Get WordCount (test run)

(d) Modify WordCount to InMapperWordCount and test run

(e) Implement **Average Computation Algorithm** to compute the average of the "last quantity" in a Apache log file for each ip address ("the first quantity).

```
64.242.88.10 - - [07/Mar/2004:16:11:58 -0800] "GET
/twiki/bin/view/TWiki/WikiSyntax HTTP/1.1" 200 7352
```

Use the data file attached.

(f) Implement the **in-mapper combining version of the Average Computation Algorithm** to compute the average of the "last quantity" in a **Apache access log** file for each ip address ("the first quantity).

```
64.242.88.10 - - [07/Mar/2004:16:11:58 -0800] "GET
/twiki/bin/view/TWiki/WikiSyntax HTTP/1.1" 200 7352
```

Apache.sample.rar has two files. Use the log file.

PART 2 - 4

Next you will create a crystal ball to predict events that may happen once a certain event happened.

Example: Amazon will say people who bought “item one” have bought the following items : “item two”, “item three”, “item four”.

For the purpose of this project you can assume that historical customer data is available in the following form. Each record contains the product IDs of all the product bought by one customer.

TEST DATA **(You must use this otherwise, you will loose 50%)**

B11 C31 A10 D76 A12 B12 C31 D76 C31 A10 B12 D76 C31 B11 // items bought by a customer, listed in the order she bought it

A10 D76 D76 B12 B11 C31 D76 B12 C31 B11 A12 C31 B12 B11 // items bought by another customer, listed in the order she bought it

…

Let the neighborhood of X, N(X) be set of all term after X and before the next X.

Example: Let Data block be [a b c a d e]

N(a) = {b, c}, N(b) = {c, a, d, e}, N(c) = {a, d, e}, N(a) ={d, e}, N(d) = {e}, N(e) = {}.

Part 2. Implement Pairs algorithm to compute relative frequencies.

a. [2 points] Create Java classes (.java files)
b. [1 points] Show input, output and batch file to execute your program at command line in Hadoop.

Part 3. Implement Stripes algorithm to compute relative frequencies.

a. [2 points] Create Java classes (.java files)
b. [1 points] Show input, output and batch file to execute your program at command line in Hadoop.

Part 4. Implement Pairs in Mapper and Stripes in Reducer to compute relative frequencies.

a. [2 points] Create Java classes (.java files)
b. [1 points] Show input, output and batch file to execute your program at command line in Hadoop.