UNIVERSITY OF ILLINOIS AT CHICAGO, INFORMATION AND DECISION SCIENCES

Frequent Itemset Mining Using MapReduce

GOAL

Using MapReduce framework to implement frequent itemsets (2-itemsets: a ⇒ b and 3 itemsets: a,b ⇒ c, where a, b, c are items).

DATA FORMAT

The original data would be a text file, each line is a transaction containing multiple items separated by commas (item1, item2, item3,$\cdots$, itemn). Let's use the following dummy data as an example. Each line is a basket or a transaction. Each number is an item id.

trans_1 1 2 5
trans_2 2 3
trans_3 2 4 5
trans_4 1 2
trans_5 1 5

IMPLEMENTATION

The mapper class:
Your map function would take this original file and generate an intermediate output. The input key would be Text: trans_i (e.g., trans_1). The input value would be Text: the rest content in each line. The output key would be Text: the singleton or doubleton itemsets. The output value is IntWritable: 1. In the example case, it will generate the following outputs.

1 2 5 ⇒ (1,1), (2,1), (5,1), ((1,2),1), ((1,5),1), ((2,5),1), ((1,2,5),1)
2 3 ⇒ (2,1), (3,1), ((2,3),1)
2 4 5 ⇒ (2,1), (4,1), (5,1), ((2,4),1), ((2,5),1), ((4,5),1), ((2,4,5),1)
1 2 ⇒ (1,1), (2,1), ((1,2),1)
1 5 ⇒ (1,1), (5,1), ((1,5),1)

The reducer class:
Your reduce function would aggregate all values for each key. The output key would be Text: itemsets. The output value is IntWritable: the number of occurrence of each corresponding key. In the example, case, it will generate the following outputs: output-1.
output-1:
(1,3),(2,4),(3,1),(4,1),(5,3),((1,2),2),((1,5),2),((2,3),1),((2,4),1),((2,5),2),((4,5),1)
((1,2,5),1), ((2,4,5),1)

The partition class:
Your partition class should extend the existing Partition class in Hadoop. You have to overwrite the getPartition(key, value, numberReduceTasks). Within your driver class, you have to make sure the number of reduce tasks is the same as the number of partition you split your data into in the partition function.

Driver class
For the Mapper, you need to use KeyValueTextInputFormat to separate key-value pairs for each line instead of using default TextInputFormat (key: LongWritable, value: the content of the line).

SUBMISSION

You need to submit one zip file, which contain all source codes (*.java files) and README file. The name of your zip file would be following the format like: Firstname_Lastname_Assignment2.zip. Please follow this format and submit it through the blackboard. Please try to comment your codes for critical sections and make your codes as readable as possible.