Nan Li
CSE 6242 HW3 Q1
March 31, 2017

Analysis a Graph with Hadoop/Java

(a):

Step 1: Load data and set up configurations for MapReduce jobs.

Step 2: Preprocess data by splitting each line of cvs file into a list of strings. The output is like:

```
117 51 1

194 51 1

299 51 3

230 151 51

194 151 79

51 130 10
```

Step 3: loop through each string list, select the second string as key, third string weight as value. After the mapping, we generate key-value pairs representing the relation between a node and its inbound weight. The output is as follows:

<51 1>
<51 1>
<51 3>
<151 51>
<151 79>
<130 10>

Step 4: Reduce by key. The input is as follows:

<51 [1,1,3]>
<151 [51,79]>
<130 10>

For key-value pairs with same keys, iterate through all values to find the maximum value, and save it as value for key. The output is as follows:

<51 3>
<151 79>
<130 10>

Step 5: Specifying the schema and save it into file system

(b)Basically we use the idea from http://codingjunkie.net/mapreduce-reduce-joins/. In mapping process, we will implement secondary sorting by tagging the key with either "1" or "2". So values with key tagging 1 will come before that with "2", in this case student name will come before department name. Then we partition and group data by joinKey(Department_ID) for reduce job. These mapping result will be sent to Reducer. We iterate through all records, append student info and department info which act as the value for joinKey. Finally, process the data into acceptable file format to file system.

Step 1: Load data and preprocessing.

Step 2: Set Department_ID as key, the remaining string as value. The output is as follows:

<1234 (Student, Alice)>

<1234 (Student, Bob)>

<1123 (Department, CSE)>

<1234 (Department, CS)>

<1123 (Student, Joe)>

Step 3: Reduce by key. The input is as follows:

<1234 [(Student, Alice), (Student, Bob)] (Department, CS)>

<1123 (Student, Joe)(Department, CSE)>

The output is as follows:

<1234 Student Alice Department CS>

<1234 Student Bob Department CS>

<1123 Student Joe Department CSE>

Step 4: Sort records with department_id ascending and Student name descending. Save the key-value pairs into file system by specifying the schema. The output is as follows

<1123, Joe, CSE>

<1234, Bob, CS>

<1234, Alice, CS>