Comparisons of runtimes - Assignment 3 Report

Neelesh and Habiba

Implementation details for the MapReduce:

Shell script

We created a shell script which invokes calls to 1. Java program of Sequential version 2. Java program of Threaded version 3. Hadoop program for MapReduce Pseudo mode 4. EMR commandline call to create a cluster and run the step.

We run these commands for each of the following options 1. Mean (Option 0) 2. Median (Option 1) 3. Fast Median (Option 2)

Inputs

Input1 - The .gz files provided in the all folder are placed in /input/input1 folder Input2 - some of the files of the Input1 are placed in /input/input2 folder

Output from shell script

Output files from the programs will be written as below to /output/

```
    seq_fastmedian_input1.txt
    seq_fastmedian_input2.txt
    seq_mean_input1.txt
    seq_mean_input2.txt
    seq_median_input1.txt
    seq_median_input2.txt
    threaded_fastmedian_input1.txt
    threaded_fastmedian_input2.txt
    threaded_mean_input1.txt
    threaded_mean_input2.txt
    threaded_median_input1.txt
    threaded_median_input2.txt
    ...
```

Output file from the shell script: runtimes.csv

Sample Output from shell script

The output file runtimes.csv will contain rows as below:

```
sequential,input1,mean,121.389
sequential,input1,median,132.032
sequential,input1,fastmedian,123.884
sequential,input2,mean,55.089
sequential,input2,median,55.128
sequential,input2,fastmedian,54.677
threaded,input1,mean,70.010
threaded,input1,median,72.563
....
```

The first column is the airline,second is average ticket price per month, third column is Month number and the final is count of flights the airline runs in total. This data is redundantly returned by the reducer.

Running the shell script

Prerequisites

Input files should be placed in input/input1 and input/input2

```
# Execute the script in the following way
./benchmark-script
```

Conclusion

From the results obtained by 2 input sizes, it seems that threaded version of the program runs quite faster than the other versions. But if we run the program for data which is too big, fully distributed version of the program(on emr) would be more efficient. Values and Fast Median and Median are 99% accurate. The runtime calculate only fast median is O(n) where as to calculate actual median it takes O(nlogn) as it involves sorting.

Results - Plot

Following graph shows time taken for each configuration to run with different input sizes: