**Running instructions:**

This file contains data on how to run each of the four tasks using the standalone jar in Hadoop.

1. Upload the three text files ("Melbourne", "RMIT", "3littlepigs") into one folder in HUE, called "input"
2. Upload the jar file in HUE then copy jar from HDFS to Hadoop using the command line "hadoop fs –copyToLocal <HDFS-jar-file> <local-path>". Replace <HDFS-jar-file> and <local-path> with the correct details
3. To run each task using the standalone jar in Hadoop, use the below command lines for the task you wish to run (assuming the user name where the input folder lies is “s3759621”)
   1. For Task 1, run: "hadoop jar Assignment1-1.0-SNAPSHOT.jar edu.rmit.cosc2633.s3759621.Assignment1.WordLength /user/s3759621/input /user/s3759621/output1"
   2. For Task 2, run: "hadoop jar Assignment1-1.0-SNAPSHOT.jar edu.rmit.cosc2633.s3759621.Assignment1.WordFirstChar /user/s3759621/input /user/s3759621/output2"
   3. For Task 3, run: "hadoop jar Assignment1-1.0-SNAPSHOT.jar edu.rmit.cosc2633.s3759621.Assignment1.WordCount /user/s3759621/input /user/s3759621/output3"
   4. For Task 4, run: "hadoop jar Assignment1-1.0-SNAPSHOT.jar edu.rmit.cosc2633.s3759621.Assignment1.WordLengthExtended /user/s3759621/input /user/s3759621/output4"

**Requirement (g): Performance analysis**

The table below outlines the total CPU time used by the Map and Reduce tasks when the number of nodes in the cluster is raised from 3 to 5 and 7.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No. of nodes** | **CPU\_MILLISECONDS MAP Task 1** | **CPU\_MILLISECONDS REDUCE Task 1** | **CPU\_MILLISECONDS REDUCE Task 2** | **CPU\_MILLISECONDS REDUCE Task 3** |
| 3 | 251210 | 5670 | 14880 | 9570 |
| 5 | 260410 | 7780 | 14110 | 9070 |
| 7 | 259090 | 7230 | 14300 | 7300 |

Table - CPU time for task (g)

As it can be observed, the CPU time for the Map tasks does not vary to a significant degree when increasing the number of nodes. In fact, it appears that the smallest amount of time taken by the Map task happens when the number of nodes is 3 (251210 milliseconds). On the other hand, the CPU time for the Reduce tasks does appear to vary slightly. For Reduce task 1, when the number of nodes increases from 3 to 5, the CPU time increases by 2110 milliseconds, then decreases by 550 milliseconds when the number of nodes increases further to 7. For Reduce task 2 the CPU time appears to remain approximately the same when increasing the number of nodes in the cluster. However, an interesting pattern can be observed for Reduce task 3, where the CPU time decreases by a small amount of 500 milliseconds as the number of nodes increases from 3 to 5, and a further decrease of 1770 milliseconds is observed as the number of nodes increases from 5 to 7. This is an expected behaviour as more nodes in the cluster should speed up the Reduce tasks slightly.

Overall, the CPU time does not appear to vary significantly as the number of nodes are increased, although one might expect that the Map and Reduce tasks would be sped up with the increase in nodes in a cluster.