COSC 2637/2633 Big Data Processing
Assignment 1 – Tax Trip Statistics

Assessment Type	– Individual assignment. – Submit online via Canvas → Assignment 1. – Marks awarded for meeting requirements as closely as possible. – Clarifications/updates may be made via announcements or relevant discussion forums.
Due Date	Due at 23:59, 17 Aug, 2022
Marks	25

Overview

Write MapReduce programs which give your chance to develop basic understanding of principles when solving complex problems on Hadoop execution platform.

Learning Outcomes

The key course learning outcomes are:

- CLO 1: model and implement efficient big data solutions for various application areas using appropriately selected algorithms and data structures.
- CLO 2: analyse methods and algorithms, to compare and evaluate them with respect to time and space requirements and make appropriate design choices when solving real-world problems.
- CLO 3: motivate and explain trade-offs in big data processing technique design and analysis in written and oral form.
- CLO 4: explain the Big Data Fundamentals, including the evolution of Big Data, the characteristics of Big Data and the challenges introduced.
- CLO 6: apply the novel architectures and platforms introduced for Big data, i.e., Hadoop, MapReduce and Spark.

Assessment Details

You will be provided a dataset (i.e., trip) which records the kilometers of each trip of many taxis. For each taxi, count the number of trips and the average kilometers per trip by developing MapReduce programs with Python. Note using any other language like Java will lead to 0 mark of the assignment directly. Also, you are not allowed to use any python MapReduce library such as mrjob.

For example

Taxi#, km	The average kilometers per trip
00005, 20	00005, 3, 16.33
00004, 15	00001, 2, 9
00001, 11	00004, 2, 10
00005, 8	
00005, 21	
00004, 5	
00001, 7	

Task 1 – No Local Aggregation

write a MapReduce program with python to perform the task where *no local aggregation* is applied.

Task 2 – Combiner

write a MapReduce program with python to perform the task where *combiner* is applied.

Task 3 – In-mapper Combining

write a MapReduce program with python to perform the task where *in-mapper combining* is applied.

Task 4 – Performance Analysis

This task asks you to compare the codes implemented in Task 1-3 in a written report. Run the codes with 3 reducers. For each run, report

- *Map input records*
- *Map output records*
- *Combine input records*
- *Combine output records*
- *Reduce shuffle bytes*
- *Reduce input records*

Analyse the results and explain which implementation is better and why.

Submission

Your assignment should follow the requirement below and submit via Canvas > Assignment 1. Assessment declaration: when you submit work electronically, you agree to the assessment declaration:

Format Requirements

Failure to follow the requirements incurs 5 marks penalty for each.

1. If your student ID is s1234567, then please create a zip file named s1234567_BDP_A1.zip with the following files without sub-folders.
   - task1-mapper.py
   - task1-reducer.py
   - task1-run.sh
   - task2-mapper.py
   - task2-combiner.py
   - task2-reducer.py
   - task2-run.sh
   - task3-mapperInMapCombiner.py
   - task3-reducer.py
   - task3-run.sh
   - task4.pdf
   - README

2. Note shell scripts task1-run.sh, task2-run.sh, task3-run.sh are used to execute task1, task2, and task 3 respectively. Any path in the shell scripts must be specified as follows:

```
-file ./mapper.py
-mapper ./mapper.py
-file ./combiner.py
-combiner ./combiner.py
-file ./reducer.py
-reducer ./reducer.py
-input /trip.txt
-output /output
```

Functional Requirements

Failure to follow the requirements incurs up to 5 marks penalty

- The code must be well written using good coding style.
- The code must include sufficient comments which can clearly explain the major logic flow of the program.

Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e., directly copied), summarized, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to
https://www.rmit.edu.au/students/student-essentials/rights-and-responsibilities/academic-integrity

RMIT UNIVERSITY

Marking Guide

- Late submission of the assignment results in penalty of 2 marks for (up to) each 24 hours being late. Submissions more than 5*24 hours late results in zero marks.
- If unexpected circumstances affect your ability to complete the assignment, you can apply for special consideration.
  - Requests for special consideration of within 7*24 hours please can be via emailing the course coordinator directly with supporting evidence.
  - Request for special consideration of more than 7*24 hours must be via the University Special consideration: https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/assessment/special-consideration.

Task 1 Code Development	0 marks - cannot run on AWS EMR or - no/unreasonable output or - >1 major logic error in code	1 mark output incorrect due to 1 major logic error in code	2-3 marks output incorrect due to >1 minor logic error in code	4 marks output incorrect due to 1 minor logic error in code	5 marks output correct and no code error
Task 2 Code Development	0 marks - cannot run on AWS EMR or - no/unreasonable output or - >1 major logic error in code	1-2 mark output incorrect due to 1 major logic error in code	3-4 marks output incorrect due to >1 minor logic error in code	5-6 marks output incorrect due to 1 minor logic error in code	7 marks output correct and no code error
Task 3 Code Development	0 marks - cannot run on AWS EMR or - no/unreasonable output or - >1 major logic error in code	1-2 mark output incorrect due to 1 major logic error in code	3-5 marks output incorrect due to >1 minor logic error in code	6-7 marks output incorrect due to 1 minor logic error in code	8 marks output correct and no code error
Task 4 Performance Analysis –	0 marks - no/unreasonable answer	1 mark - < 50% answer correct or - two of task 1-3 output is incorrect.	2-3 marks - < 80% answer correct or - one of task 1-3 output is incorrect	4 marks correctly but verbose, clear, well written, thorough, complete, etc.	5 marks correctly and concisely
Functional requirement	Failure penalty on functional requirements detailed in specification				
Format requirement	Failure penalty on format requirements detailed in specification				