Big Data Processing
COSC 2637/2633
Assignment 1

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	Week 7, Friday 13rd September 2020, 23:59
Marks	40

1. Overview

Write MapReduce programs which gives your chance to understand the complexity of MapReduce programing, the essential components you learned in lectures, the unique debugging method, the impact of performance using different size clusters.

2. 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. Analyze met respect to time and space requirements l-world problems.

CLO 3. Motivate and gn and analysis in written and oral form.

CLO 4. Explain the Big Data Fundamentals, including t the characteristics of Big Data and the challenges introduced.

CLO 5. Apply non-relational databases, the techniques large volumes of structured and unstructured data, as well as streaming data.

CLO 6. Apply the novel architectures and platforms introduced for Big data, in particular Hadoop and MapReduce.

3. Assessment details

In Task 2 of Lab 3 (week 4), you have developed a MapReduce program and run it in Hadoop. It is the basic version of word count. In this assignment, you are asked to extend the functions based on the MapReduce program using what you learned in this course.

You should use Java to develop your MapReduce program over AWS EMR (if you want to use other code language, please contact lecturer for approval).

Task 1 – Count words by lengths (8 marks)

Write a MapReduce program to count number of short words (1-4 letters), medium words (5-7 letters) words, long words (8-10 letters) and extra-long words (More than 10 letters).

Task 2 – Count words by the first character (8 marks)

Write a MapReduce program that outputs a count of all words that begin with a vowel and count of all how many words that begin with a consonant.

Task 3 – Count word with in-mapper combining (12 marks)
Write a MapReduce program to count the number of each word where the in-mapper combining is implemented rather than an independent combiner.

Task 4 – Count word with partitioner (12 marks)
Extend the MapReduce code in Task 1 by using partitioner such that
- short words (1-4 letters) and extra-long words (More than 10 letters) are processed in one reducer,
- medium words (5-7 letters) and long words (8-10 letters) are processed in another reducer.

4. 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: https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/assessment/assessment-declaration

5. Requirement
(a) The codes for all four tasks are entailed in a single Maven project. (2 marks)
(b) Submit the complete Maven project source code in a .zip file (including a standalone jar file). The zip file should be named as sxxxxx_BDP_S2_2020.zip (replace sxxxxx by your student ID). (2 marks)
(c) You need include a "README" file in the zip file. In the README, you are asked to specify how to run each task using the standalone jar in Hadoop. (1 mark)
(d) Paths of input file and output file should not be hard-coded. (1 mark)
(e) For all tasks, use e input files and process them together in must be stored in /user/sxxxxx/out er/sxxxxx/output2 for Task 2, and so on). (4x1 marks)
(f) For each task, using Apache log4j log information:
   - In the MAP tasks, the log should be "The mapp <student ID>"
   - In the REDUCE tasks, the log should be "The reducer task of <Your Name>, <student ID>"
(g) Conduct performance analysis on different numbers of nodes in EMR clusters. To this end, run the code in Task 1 to process a large data set
   s3a://commoncrawl/crawl-data/CC-MAIN-2018-17/segments/1524125936833.6
   when the number of nodes in EMR clusters is 3, 5, 7 respectively. Show the CPU_MILLISECONDS for each MAP task and each REDUCE task in the README file (the same one mentioned in (c)); and analyze what you observed (250-500 words). (3 marks)
(h) Your MapReduce program(s) must be well written, using good coding style and including appropriate use of comments. (4x2 marks)

6. Marking Guide
(a) If one task cannot be run using the submitted jar file, no mark for this task.
(b) If one task can run but the output is incorrect. At least half mark will be deducted for this task. If the code has major issues (such as logically incorrect), 0 mark for this task.

7. 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 behaviors, 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