Big Data Processing
COSC 2637/2633
Assignment 4 – HDFS Monitoring via Spark Streaming

Assessment Type	– Individual assignment. – Submit online via Canvas → Assignment 4. – Marks awarded for meeting requirements as closely as possible. – Clarifications/updates may be made via announcements or relevant discussion forums.
Due Date	At 23:59, 2 Nov, 2022
Marks	25

Overview

Write Spark programs which gives your chance to apply the essential components you learned in lectures and labs to understand the complexity of Spark programming.

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 4 - Explain the Big Data Fundamentals, including the evolution of Big Data, the characteristics of Big Data and the challenges introduced.
- CLO 5 - Apply non-relational databases, the techniques for storing and processing large volumes of structured and unstructured data, as well as streaming data.
- CLO 6 - Apply the novel architectures and platforms introduced for Big data, i.e., Hadoop, MapReduce and Spark.

Task – Spark Streaming

Develop a spark streaming program with Scala to monitor a folder on HDFS in real-time such that any new file in the folder will be processed (the batch interval is 5 seconds). The following three tasks are implemented in the same Scala object:

A. For each RDD of Dstream, count the word frequency and save the output on HDFS. Use regular expression to make sure that each word consists of characters only (tip: `findAllIn()`). (5 marks)

B. For each RDD of Dstream, filter out the short words (i.e., < 5characters) and then count the co-occurrence frequency of words (the words are considered co-occurred if they are in the same line); save the output on HDFS. (10 marks)

C. For the Dstream, filter out the short words (i.e., < 5 characters) and then count the co-occurrence frequency of words (the words are considered co-occurred if they are in the same line); save the output on HDFS. Note you are required to use `updateStateByKey` operation to continuously update the co-occurrence frequency of words with new information. (10 marks)

Format Requirements:

Failure to follow the requirements incurs up to 8 marks penalty

(a) The source codes of three tasks are entailed in submission.
(b) Submit the developed Scala project in a single .zip file with a jar file.
(c) The zip file should be named as sxxxxx_BDP_A4.zip (replace sxxxxx by student ID).
(d) You need include a "README" file in the zip file.
(e) In README, you must specify exactly how to run the jar in AWS EMR platform.
(f) Paths of input and output should not be hard-coded.

Functional Requirements:

Failure to follow the requirements incurs up to 5 marks penalty

(a) For each task, the output on HDFS should be named with a unique sequence number. For example, taskA-001, taskA-002, taskB-001, taskB-002, taskC-001, taskC-002.
(b) You need create a single Scala project including all three tasks so that they work on the same stream data.

Submission

Your assignment should follow the requirement below and submit via Canvas > Assignment 4.

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

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

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 a Implementation Correctness	0 marks cannot operate, no output, >1 major logic error in code	1 mark 1 major logic error in code	2 marks >1 minor logic error in code	3-4 marks output incorrect due to 1 minor logic error in code	5 marks output correct and no code error
Task b Implementation Correctness	0 marks cannot operate, no output, >1 major logic error in code	1-3 marks 1 major logic error in code	4-6 marks >1 minor logic error in code	7-9 marks output incorrect due to non-logic errors or 1 minor logic error in code	10 marks output correct and no code error
Task c Implementation Correctness	0 marks cannot operate, no output, >1 major logic error in code	1-3 marks 1 major logic error in code	4-6 marks >1 minor logic error in code	7-9 marks output incorrect due to non-logic errors or 1 minor logic error in code	10 marks output correct and no code error
Functional requirement	Penalty applied for not following the functional requirements (detailed in specification above)				
Format requirement	Penalty applied for not following the format requirements (detailed in specification above)				