

## COSC 2637/2633 Big Data Processing

### Assignment 2 - Parallel Breadth-First Search on Large Graphs

Assessment Type	<ul style="list-style-type: none"><li>– Individual assignment.</li><li>– Submit online via Canvas → Assignment 2.</li><li>– Marks awarded for meeting requirements as closely as possible.</li><li>– Clarifications/updates may be made via announcements or relevant discussion forums.</li></ul>
Due Date	At 23:59, 5 Oct, 2022
Marks	25

#### Overview

Write an advanced MapReduce program which gives your chance to develop in-depth understanding of principles when solving complex problems on Hadoop execution platform and analyse solutions by applying the knowledge learned in this course to achieve the optimal outcome.

#### 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

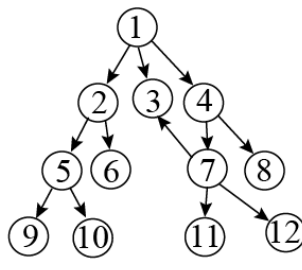
Given the source node  $s$  in a graph  $G$ , an important task is to find the shortest path distance from  $s$  to all other nodes in  $G$ . This course introduces a solution known as *Parallel Breadth-First Search* (BFS) in details.

#### Task 1 – Code Development

write a MapReduce program with python to implement BFS. Table-1 shows an example graph and its representation in graph.txt (where *none* means the corresponding node has no out-link neighbours). The distance.txt indicates the distance from source node to each node after initialization (the node #1 is the source, the distance from source to node #1 is 0, to other nodes is 9999).

Your code must represent graph and initialize the distance in the same way as in Table-1. Also, you must implement the code following the framework introduced in the lab for the k-means clustering. Failure to do so leads to 0 marks of assignment. Also, you are not allowed to use any python MapReduce library such as mrjob.

**Table-1 – Graph representation and initialization.**

Graph	graph.txt node#: a list of out-link neighbours	distance.txt node#: the initial distance from source
	1: 2 3 4 2: 5 6 3: none 4: 7 8 5: 9 10 6: none 7: 3 11 12 8: none 9: none 10: none 11: none 12: none	1: 0 2: 9999 3: 9999 4: 9999 5: 9999 6: 9999 7: 9999 8: 9999 9: 9999 10: 9999 11: 9999 12: 9999

The expected output of the example in Table-1 is

distance.txt node#: the shortest path distance from source
1: 0 10: 3 11: 3 12: 3 2: 1 3: 1 4: 1 5: 2 6: 2 7: 2 8: 2 9: 3

### Task 2 - Performance Analysis

Suppose you have a very large graph with millions of nodes such as a road network or social networks. In the framework introduced in the lab for the k-means clustering, the number of reducer task is set to 1. What is the disadvantage of this setting? What is your solution to address this disadvantage? Note your solution must be detailed and complete other than a high-level description. Write a report in a PDF file.

### Submission

Your assignment should follow the requirement below and submit via Canvas > Assignment 2. Assessment declaration: when you submit work electronically, you agree to the [assessment declaration](#):

### Format Requirements

Failure to follow the requirements incurs up to 6 marks penalty

- If your student ID is s1234567, then please create a zip file named s1234567\_BDP\_A2.zip with the following files without sub-folders.
  - a. All Python files you have developed.
  - b. run.sh: a bash script to run your MapReduce job on the EMR master node.
  - c. report.pdf: a PDF file for task 2.
  - d. README: a text file that includes your student's name, student ID, and how to run your code.
- Do NOT submit the Hadoop Streaming jar file.
- Do NOT submit the given input files.
- Any path in the shell scripts must be specified as follows:
  - -file ./distances.txt
  - -file ./mapper.py
  - -mapper ./mapper.py
  - -file ./reducer.py
  - -reducer ./reducer.py
  - -input /graph.txt
  - -output /output
- Please assume the Hadoop Streaming jar file and all your Python files are in the same folder on the EMR master node.

### 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>

### 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 - 1 Code Development – mapper	0 marks - cannot run on AWS EMR or - no/unreasonable output or - not follow input format 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 1 - 2 Code Development - reducer	0 marks - cannot run on AWS EMR or - no/unreasonable output or - not follow input format or - >1 major logic error in code	1-3marks output incorrect due to 1 major logic error in code	4-6 marks output incorrect due to >1 minor logic error in code	7-9 marks output incorrect due to 1 minor logic error in code	10 marks output correct and no code error
Task 1 - 3 Code Development – iteration	0 marks - cannot run on AWS EMR or - no/unreasonable output or - not follow input format or - >1 major logic error in code	1 mark Logic incorrect due to 1 major logic error in code	2-3 marks Logic incorrect due to >1 minor logic error in code	4 marks Logic incorrect due to 1 minor logic error in code	5 marks Logic correct and no code error
Task 2 Performance Analysis – analysis report	0 marks - < 10 marks in Task 1 - no/unreasonable report “what is the performance bottleneck”.	1 mark answer partially correct with major issues including logic error(s) or incorrect statement(s) in report	2-3 marks answer partially correct with minor issues, e.g., no obvious logic error but with improper statement(s) in report.	4 marks answer correctly in general but verbose, e.g., 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				