Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code: CMT 202
Module Title: Distributed and Cloud Computing
Lecturer: Padraig Corcoran
Assessment Title: CMT 202 Coursework 1
Assessment Number: 1
Date Set: Monday 8 March 2021.
Submission Date and Time: Monday 12 April 2021 at 9:30am.
Return Date: by Monday 10 May 2021.

This assignment is worth 25 % of the total marks available for this module. If coursework is submitted late (and where there are no extenuating circumstances):

1 If the assessment is submitted no later than 24 hours after the deadline, the mark for the assessment will be capped at the minimum pass mark;

2 If the assessment is submitted more than 24 hours after the deadline, a mark of 0 will be given for the assessment.

Your submission must include the official Coursework Submission Cover sheet, which can be found here:
https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf

Submission Instructions

All submission should be via Learning Central unless agreed in advance with the Director of Teaching.

Description		Type	Name
Cover sheet	**Compulsory**	One PDF (.pdf) file	[student number].pdf
Solutions	**Compulsory**	One zip (.zip) file containing the Python code developed.	[student number].zip

Any code submitted will be run on a university provided laptop. The only additional Python library, other than those already installed on the university provided laptop, which will be used to run this code is mrjob.

Any deviation from the submission instructions above (including the number and types of files submitted) may result in a mark of zero for the assessment or question part.

Staff reserve the right to invite students to a meeting to discuss coursework submissions.

Assignment

This coursework requires you to write four MapReduce programs. These programs should be written using Python 3 and the Python mrjob library. Each solution should distribute computation across multiple map and/or reducer tasks.

Part 1

Given a CSV file where each line contains a set of numbers, write a MapReduce program which determines the maximum of all numbers in the file. For example, consider the following sample CSV file:

```
2,2,3
4,3
```

Given this CSV file, the maximum is 4.

Entitle the python program in question part1.py. That is, entering the following command at the terminal should result in your MapReduce program being applied to fileName.csv

```
pipenv run python part1.py fileName.csv
```

Part 2

Write a mapReduce program which takes as input a CSV file containing comma separated words and outputs for each word the lines that the word appears in. For example, consider the following file:

```
goat,chicken,horse
cat,horse
dog,cat,sheep
buffalo,dolphin,cat
sheep
```

The corresponding output will be the following:

```
"buffalo"   ["buffalo,dolphin,cat"]
"cat"        ["buffalo,dolphin,cat", "cat,horse", "dog,cat,sheep"]
"chicken"   ["goat,chicken,horse"]
"dog"        ["dog,cat,sheep"]
"dolphin"   ["buffalo,dolphin,cat"]
"goat"       ["goat,chicken,horse"]
"horse"      ["cat,horse", "goat,chicken,horse"]
"sheep"      ["dog,cat,sheep", "sheep"]
```

Entitle the python program in question part2.py. That is, entering the following command at the terminal should result in your MapReduce program being applied to fileName.csv

```
pipenv run python part2.py fileName.csv
```

Part 3

Given a file containing words separated by spaces, write a MapReduce program which counts the number of times each 4 word sequence appears in the file.

For example, consider the following file:

```
one two three four seven one two three four
three four seven one
seven one two three
```

The number of times each 4 word sequence appears in this file is:

```
"three four seven one"    2
"four seven one two"       1
"one two three four"       2
"seven one two three"     2
"two three four seven"     1
```

Entitle the python program in question part3.py. That is, entering the following command at the terminal should result in your MapReduce program being applied to fileName.txt

```
pipenv run python part3.py fileName.txt
```

Part 4

Uniform Resource Locator (URL) links describe the structure of the web. Consider a CSV file where each line contains two URLs which specify a single link. That is, the first and second values on each line specify the source and destination of the link in question. For example, consider the following sample CSV file:

```
url1,url2
url1,url3
url2,url3
url4,url5
url2,url4
```

Given such a CSV file, write a MapReduce program which finds all paths of length two in the corresponding URL links. That is, it finds the triples of URLs (u, v, w) such that there is a link from u to v and a link from v to w.

For example, the sample CSV file above contains the following paths of length two:

```
url2, url4, url5
url1, url2, url3
url1, url2, url4
```

Entitle the python program in question part4.py. That is, entering the following command at the terminal should result in your MapReduce program being applied to fileName.csv

```
pipenv run python part4.py fileName.csv
```

Learning Outcomes Assessed

The following learning outcomes from the module description are specifically being assessed in this assignment:
Demonstrate and apply knowledge about the state-of-the-art in distributed-systems architectures.
Understand issues in distributing an application across a network.

Criteria for assessment

Credit will be awarded against the following criteria.

Marks will be assigned to each of the four parts specified above as follows:
Successfully implement part 1 specified above. [6 marks]
Successfully implement part 2 specified above. [6 marks]
Successfully implement part 3 specified above. [6 marks]
Successfully implement part 4 specified above. [7 marks]

The quality of your solution for each part will be determined based on its performance on a corresponding set of test cases. Feedback on your performance will address each of these criteria.

A student can expect to receive a distinction (70-100%) if they correctly implement all parts.
A student can expect to receive a merit (60-69%) if they correctly implement most parts with only minor errors.
A student can expect to receive a pass (50-59%) if they correctly implement some parts without major errors.
A student can expect to receive a fail (0-50%) if they fail to correctly implement some parts without major errors.

IMPORTANT – All code submitted must be written in Python 3 and use the mrjob library to implement MapReduce operations.

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned on Monday 10 May 2021 via Learning Central. Where requested, this will be supplemented with oral feedback.

Feedback from this assignment will be useful for the second coursework in this module.