CS1003 Practical 5: Data Processing with MapReduce

This practical is worth 25% of the overall coursework mark—all five practicals are equally weighted, and your lowest mark will be ignored. You **must submit an acceptable attempt at all five practicals**; failure to do so may result in failing the module without right to reassessment.

As for every practical, you should arrive in the lab having prepared in advance, by studying the practical handout and reading up on any relevant lecture, tutorial or exercise. If you don't do this you will waste time during your lab sessions.

There are no hints for this practical, since you should have seen enough examples of MapReduce in lectures and exercises.

Skills and Competencies

Competency:

- using MapReduce to process data

Necessary skills:

- expressing an algorithm in the MapReduce style
- choosing appropriate classes and methods from the MapReduce API
- identifying and dealing with possible error conditions
- testing and debugging
- writing clear, tidy, consistent and understandable code

Requirements

The practical involves manipulating fairly large data files using the Hadoop implementation of MapReduce.

In the following directory on studres there are several files containing tweets sent on one day during the 2012 Olympics[1]:

https://studres.cs.st-andrews.ac.uk/CS1003-PD/Practicals/P05/twitter-data/data-split/

Since the files are large, there is also a local copy of the directory on each iMac:

/usr/local/cs/CS1003-PD/Practicals/P05/twitter-data/

When working in the lab, please use the files from here rather than copying from studres. If you need to open the directory in the Finder you can do so using **Go > Go to Folder...**

Your program should use Hadoop to create a separate output file for each of the following:

- all tweets containing a given substring within the text of the tweet
- all tweets sent within a given time period
- the top 3 users most frequently mentioned within the text of a tweet, with the total number for each

You can reuse or adapt any of the example code provided in lectures and exercises. **Hint**: if you need to pass a parameter value to a mapper or reducer, you can do so by calling the method:

void set(String property_name, String value)

on the *JobConf* instance, and overriding the method:

void configure(JobConf job)

in the mapper or reducer class. Inside the *configure* method you can access the parameter value by calling the method:

String get(String property_name)

[1] Courtesy of Dr Alex Voss. Please read the 'READ_ME' file for conditions of use.

As usual, your program should deal gracefully with possible errors. The source code for your program should follow common style guidelines, including:

- formatting code neatly
- consistency in name formats for methods, fields, variables
- avoiding embedded "magic numbers" and string literals
- minimising code duplication
- avoiding long methods
- using comments and informative method/variable names to make the code obvious to the reader

Deliverables

Hand in via MMS, by the specified deadline on Monday of Week 12:

- Your complete Java project.
- A brief report (maximum 2 pages) explaining the decisions you made, how you tested your program, and how you solved any difficulties that you encountered. Also give one piece of feedback from your Practical 4 submission, and say how you used it to improve this submission.
- You can use any software you like to write your report, but your submitted version must be in PDF format.

Extensions

If you wish to experiment further, you could try any or all of the following:

- Generate web pages to display the results
- Refine your solution so that it works with the files in the directory *data-single-line*; these contain the original files in which the data is all on one line
- Find the 5-minute period in which the most tweets were issued
- Plot the location of tweets on a map

Marking

See the standard mark descriptors in the School Student Handbook:

https://studres.cs.st-andrews.ac.uk/Library/Handbook/academic.html#Mark_Descriptors

For this practical, full marks can be obtained with a good solution to the main problem.

The standard penalty for late submission applies (1% per hour late).