School of Computing and Information Technology
University of Wollongong

Session: **Autumn 2022**
Lecturer: **Janusz R. Getta**

ISIT912 Big Data Management
Assignment 1
Published on 25 July 2022

Scope

The objectives of Assignment 1 include implementation of HDFS application and implementation MapReduce applications.

This assignment is due on **Saturday, 20 August 2021, 7:00pm** (sharp).

This assignment is worth **15%** of the total evaluation in the subject.

The assignment consists of 5 tasks and specification of each task starts from a new page.

Only electronic submission through Moodle at:
`https://moodle.uowplatform.edu.au/login/index.php`
will be accepted. A submission procedure is explained at the end of Assignment 1 specification.

A policy regarding late submissions is included in the subject outline.

Only one submission of Assignment 1 is allowed and only one submission per student is accepted.

A submission marked by Moodle as "late" is always treated as a late submission no matter how many seconds it is late.

A submission that contains an incorrect file attached is treated as a correct submission with all consequences coming from the evaluation of the file attached.

All files left on Moodle in a state "`Draft(not submitted)`" will not be evaluated.

A submission of compressed files (zipped, gzipped, rared, tared, 7-zipped, lhzed, … etc) is not allowed. The compressed files will not be evaluated.

An implementation that does not compile well due to one or more syntactical and/or run time errors scores no marks.

The first assignment is an **individual assignment** and it is expected that all its tasks will be solved **individually without any cooperation** with the other students. However, it is allowed to declare in the submission comments that a particular component or task of this assignment has been implemented in cooperation with another student. In such a case evaluation of a task or component may be shared with another student. In all other cases plagiarism will result in a **FAIL** grade being recorded for entire assignment. If you have any doubts, questions, etc. please consult your lecturer or tutor during laboratory/tutorial classes or over e-mail.

Task 1 (1 marks)
Moving a file in HDFS

Read an analyse HDFS applications provided in the files `FileSystemCat.java` and `FileSystemPut.java` and available in a folder `Resources` attached to a specification of laboratory class for `Week2` on Moodle.

Use the applications `FileSystemCat.java` and `FileSystemPut.java` to implement in Java HDFS application, that moves a file from one location in HDFS into another location in HDSF.

The application must have the following two parameters.
(1) A path to and a name of file to be moved from.
(2) A path to and a new name of file to be moved to.

Perform the following steps.

Implement the application and save its source code in a file `Solution1.java`.

Compile the Java source code and create a `jar` file.

Upload to HDFS a small text file for the purpose of future testing. A name and location of the file in HDFS is up to you.

Use Hadoop to process your application that moves a file on HDFS from one location to the other.

Use Hadoop to provide an evidence that the file earlier uploaded to HDSF has been successfully moved.

Deliverables
A file `solution1.pdf` that contains a listing of source code of your application, a report from compilation, creation of jar file, uploading to HDFS a small file for testing, processing of the application and an evidence that the file has been moved to a new location in HDFS.

Task 2 (2 marks)
Implementation of a simple MapReduce application

Read an analyse MapReduce application provided in a file `Filter.java` available in a folder `Resources` attached to a specification of laboratory class for `Week3` on Moodle.

The application has the functionality equivalent to the functionality of the following SQL statement:

```
SELECT key, value
FROM sequence-of-key-value-pairs
WHERE value > given-value;
```

An objective of this task is to use the Java code provided in a file `Filter.java` to implement a MapReduce application `Solution2` that has the functionality equivalent to the functionality of the following SQL statement:

就是我运行了Filter.java这个程序后，value-pairs
WHERE value > given-value变成了WHERE value BETWEEN given-value-1 AND given-value-2

```
SELECT key, value
FROM sequence-of-key-value-pairs
WHERE value BETWEEN given-value-1 AND given-value-2;
```


Save your solution in a file `Solution2.java`.

An input data set with `sequence-of-key-value-pairs` pairs is up to you.

When ready list, compile, create `jar` file, and process the application. Display an input dataset with `sequence-of-key-value-pairs` and the results created by the application. When completed, Copy and Paste all messages from a Terminal screen into a file `solution2.pdf`.

Deliverables
A file `solution2.pdf` with a report from displaying input data set with `sequence-of-key-value-pairs`, listing of source code of your application, report from compilation, creating `jar` file, processing the application, and displaying the results of processing of MapReduce application `Solution2`.

Task 3 (3 marks)
Implementation of a simple MapReduce application

Read an analyse MapReduce application provided in a file `MinMax.java` available in a folder `Resources` attached to a specification of laboratory class for `Week3` on Moodle.

The application has the functionality equivalent to the functionality of the following SQL statement.

```
SELECT key, MIN(value), MAX(value)
FROM sequence-of-key-value-pairs
GROUP BY key;
```

An objective of this task is to use the Java code provided in a file `MinMax.java` to implement a MapReduce application `Solution3` that has the functionality equivalent to the functionality of the following SQL statement.

```
SELECT key, AVG(value)
FROM sequence-of-key-value-pairs
GROUP BY key;
```


Save your solution in a file `Solution3.java`.

An input data set with `sequence-of-key-value-pairs` pairs is up to you.

When ready list, compile, create `jar` file, and process the application. Display an input dataset with `sequence-of-key-value-pairs` and the results created by the application. When completed, Copy and Paste all messages from a Terminal screen into a file `solution3.pdf`.

Deliverables
A file `solution3.pdf` with a report from displaying input data set with `sequence-of-key-value-pairs`, listing of source code of your application, report from compilation, creating `jar` file, processing, and displaying the results of processing of MapReduce application `Solution3`.

Task 4 (4 marks)
Implementation of MapReduce application

Read an analyse MapReduce application provided in a file `WordCount.java` available in a folder `Resources` attached to a specification of laboratory class for `Week3` on Moodle.

The application counts the total number of occurrences of words in a given text.

Assume the following classification of words depending on the length of each word.

```
very short:  1 <= length <= 3
short:       4 <= length <= 5
medium:      6 <= length <= 8
long:        9 <= length <= 12
X long:     13 <= length <= 15
XX long:    16 <= length
```

Extend Java code of the application such that it counts in a given text the total number of words in each category. For example, distribution of words in a text that consists of `90` words could be the following.


```
X short:     20 words
short:       15 words
medium:      35 words
long:        20 words
X long:      10 words
XX long:      0 words
```

Save your solution in a file `solution4.java`.

A file with input data set is up to you.

When ready, list a file with input data, list source code of your application, compile your application, create `jar` file, and process your application. Display the results created by the application. When finished, Copy and Paste the messages from a Terminal screen into a file `solution4.pdf`.

Deliverables
A file `solution4.pdf` with a listing of input data, listing of source code of the application, a report from compilation, creating `jar` file, processing, and displaying the results of processing `solution4.pdf`.

Task 5 (5 marks)
Describing MapReduce implementation

Assume, that a very large text file `crime-stories.txt` contain the texts of large number of crime stories. Assume, that the file is formatted such that one statement is located in one line of the text file.

Assume, that a small text file `patterns.txt` contains the text patterns, for example regular expressions. Assume, that the file is formatted, such that one pattern is located in one line of the text file.

To simplify the problem, assume that all text patterns in a file `patterns.txt` are different.

Finally, assume that a function `match(text-line, text-pattern)` returns true when `text-line` matches a pattern `text-pattern`. Otherwise, the function returns false.

Your task is to explain how to implement a MapReduce application that for each text pattern in a file `patterns.txt` finds the total number of statements in a `file crime-stories.txt` that match the pattern.

You must specify the parameters (if any) of your application and the key-value data in the input and output of the Map and Reduce stages

.

There is no need to write Java code, however, if you like it then it is all right to do so. The precise explanations in plain English or in a pseudocode will do. Please note, that if you decide to use pseudocode then your explanations must precisely explain what happens at each stage of Map-Reduce application.

Save you explanations in a file `solution5.pdf`. This task does not require you to write any code in Java. However, the comprehensive explanations related to all stages of data processing are expected. You are allowed to support your explanations with the fragments of pseudocode. Try to be as specific as it is possible.

Deliverables
A file `solution5.pdf` with the comprehensive explanations how would you implement in Java a MapReduce application that for each text pattern in a file `patterns.txt` finds the total number of statements in a `file crime-stories.txt` that match the pattern.

Submission of Assignment 1

Note, that you have only one submission. So, make it absolutely sure that you submit the correct files with the correct contents. No other submission is possible !

Submit the files **`solution1.pdf`**, **`solution2.pdf`**, **`solution3.pdf`**, **`solution4.pdf`**, and **`solution5.pdf`** through Moodle in the following way:

(1) Access Moodle at **`http://moodle.uowplatform.edu.au/`**
(2) To login use a **`Login`** link located in the right upper corner the Web page or in the middle of the bottom of the Web page
(3) When logged select a site **`ISIT312/912 (S222) Big Data Management`**
(4) Scroll down to a section **`Assessment items (Assignments)`**
(5) Click at **`In this place you can submit the outcomes of your work on the tasks included in Assignment 1`** link.
(6) Click at a button **`Add Submission`**
(7) Move a file **`solution1.pdf`** into an area **`You can drag and drop files here to add them`**. You can also use a link **`Add`**...
(8) Repeat step (7) for the remaining files **`solution1.pdf`**, **`solution2.pdf`**, **`solution3.pdf`**, **`solution4.pdf`**, and **`solution5.pdf`**
(9) Click at a button **`Save changes`**
(10) Click at a button **`Submit assignment`**
(11) Click at the checkbox with a text attached: **`By checking this box, I confirm that this submission is my own work,`** ... in order to confirm authorship of your submission.
(12) Click at a button **`Continue`**

End of specification