Department of Computer Science

Summative Coursework Set Front Page

Module Title: Distributed and Parallel Computing
Module Code: CS3DP
Lecturer responsible: Dr Xiaomin Chen
Type of Assignment: Coursework
Individual / Group Assignment: Individual
Weighting of the Assignment: 50%
Total marks: 100
Page limit/Word count: 4 pages
Expected hours spent for this assignment: 20 hours

Items to be submitted: 4-page report (excluding title page, diagrams, graphs and references) in pdf format
Work to be submitted on-line via Blackboard Learn by: Monday 12th May 12 Noon
Work will be marked and returned by: Tuesday 4th June.

NOTES

By submitting this work, you are certifying that it is all your sentences, figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work except where explicitly the works of others have been acknowledged, quoted, and referenced. You understand that failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly. The University's Statement of Academic Misconduct is available on the University web pages.

If your work is submitted after the deadline, *10%* of the maximum possible mark will be deducted for *each* working day (or part of) it is late. A mark of zero will be awarded if your work is submitted more than 5 working days late. You are strongly recommended to hand work in by the deadline as a late submission on one piece of work can impact on other work.

If you believe that you have a valid reason for failing to meet a deadline then you should make an Exceptional Circumstances request and submit it *before* the deadline, or as soon as is practicable afterwards, explaining why. To make such a request log on to RISIS and on the Actions tab select Exceptional Circumstance: as explained at https://www.reading.ac.uk/essentials/The-Important-Stuff/Rules-and-regulations/Exceptional-Circumstances

1. Assessment Classifications

First Class (>= 70%)	Outstanding/excellent work with correct results, a good presentation of the theoretical concepts applied to applications. A good presentation of the code and results, and a critical analysis of the results. An outstanding work will present full solutions with an insightful discussion.
Upper Second (60-69%)	Very good work with partial correct results: most work has been carried out correctly. Some tasks have not been carried out or are not completely correct. The presentation is good, well structured, clear and complete with respect to the work done.
Lower Second (50-59%)	Good work, which is missing some significant part of the assignment, and/or with partially correct results. Some tasks have not been carried out. The presentation is, in general, accurate and complete, but it lacks clarity (presentation quality).
Third (40-49%)	Acceptable solutions to limited part of the assignment. Some tasks have not been carried out. Some results may not be complete or technically sound. The presentation is not accurate, complete and lacks clarity.
Pass (35-39%)	Partial solutions to limited part of the assignment. Some tasks have not been carried out. Some results may not be complete or technically sound. The presentation is not accurate, complete and lacks clarity.
Fail (0-34%)	Incomplete solutions to limited part of the assignment. Most tasks have not been carried out with sufficient accuracy. Results may not be correct or technically sound. The presentation is not accurate, complete and lacks clarity.

2. Assignment Description

Task 1. Understanding the CAP Theorem. [25 marks]

Choose a real-life distributed computing system, such as the distributed Domain Name Server (DNS), not one that we've introduced in the lectures. Analyse its distributed architecture and the rationale behind its design including aspects such as data storage, algorithms, and investigate how the CAP theorem is applied in the design and operations.

Analyse and discuss the following:

a) The semantical properties: concurrency, availability and durability
b) The wish-list for distributed software, consider **whether and how** the chosen system has the following properties: High-availability, Fault tolerance, scalability, Extensibility, Usability, debuggability and Efficiency.
c) How is the CAP theorem applied for the chosen distributed system and why?

Please note: your discussion should incorporate specific function/service examples provided by the system to show your understanding of the CAP theorem. A general introduction to the chosen distributed computing system and the semantical properties is not sufficient.

Task 2. Understanding the Hadoop MapReduce Framework [35 marks]

You have been provided with a dataset below containing web server log files from a large e-commerce platform. Each log entry contains:

- IP Address (e.g., 192.168.1.1)
- Timestamp (e.g., [10/Jan/2024:10:05:23 +0000])
- Requested URL (e.g., "GET /product/123 HTTP/1.1")
- HTTP Status Code (e.g., 200)
- User Agent (e.g., Mozilla/5.0)

```
192.168.1.1 - - [10/Jan/2024:10:05:23 +0000] "GET /home HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.2 - - [10/Jan/2024:10:06:12 +0000] "GET /product/123 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.3 - - [10/Jan/2024:10:07:45 +0000] "GET /cart HTTP/1.1" 302 "Mozilla/5.0"
192.168.1.1 - - [10/Jan/2024:10:08:50 +0000] "GET /home HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.4 - - [10/Jan/2024:10:10:30 +0000] "GET /checkout HTTP/1.1" 500 "Mozilla/5.0"
192.168.1.2 - - [10/Jan/2024:10:15:42 +0000] "GET /product/456 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.5 - - [10/Jan/2024:10:20:18 +0000] "GET /home HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.6 - - [10/Jan/2024:10:22:30 +0000] "GET /product/789 HTTP/1.1" 404 "Mozilla/5.0"
192.168.1.7 - - [10/Jan/2024:10:25:47 +0000] "GET /cart HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.3 - - [10/Jan/2024:10:30:59 +0000] "GET /home HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.8 - - [10/Jan/2024:10:35:10 +0000] "GET /checkout HTTP/1.1" 500 "Mozilla/5.0"
```

```
192.168.1.2 - - [10/Jan/2024:10:40:25 +0000] "GET /product/123 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.2 - - [10/Jan/2024:10:42:47 +0000] "GET /cart HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.4 - - [10/Jan/2024:10:43:45 +0000] "GET /product/357 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.9 - - [10/Jan/2024:10:45:33 +0000] "GET /product/123 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.2 - - [10/Jan/2024:10:46:30 +0000] "GET /checkout HTTP/1.1" 500 "Mozilla/5.0"
192.168.1.5 - - [10/Jan/2024:10:46:40 +0000] "GET /product/235  HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.5 - - [10/Jan/2024:10:50:45 +0000] "GET /product/357 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.9 - - [10/Jan/2024:10:51:40 +0000] "GET /product/246  HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.5 - - [10/Jan/2024:10:52:47 +0000] "GET /cart HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.5 - - [10/Jan/2024:10:52:52 +0000] "GET /product/579 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.9 - - [10/Jan/2024:10:53:50 +0000] "GET /product/135 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.5 - - [10/Jan/2024:10:55:50 +0000] "GET /checkout HTTP/1.1" 500 "Mozilla/5.0"
192.168.1.10 - - [10/Jan/2024:11:00:12 +0000] "GET /contact HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.6 - - [10/Jan/2024:11:05:29 +0000] "GET /product/456 HTTP/1.1" 404 "Mozilla/5.0"
192.168.1.2 - - [10/Jan/2024:11:15:00 +0000] "GET /product/123 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.5 - - [10/Jan/2024:11:35:10 +0000] "GET /checkout HTTP/1.1" 500 "Mozilla/5.0"
192.168.1.8 - - [10/Jan/2024:11:40:42 +0000] "GET /product/789 HTTP/1.1" 404 "Mozilla/5.0"
192.168.1.9 - - [10/Jan/2024:11:50:30 +0000] "GET /home HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.10 - - [10/Jan/2024:12:00:05 +0000] "GET /checkout HTTP/1.1" 500 "Mozilla/5.0"
192.168.1.11 - - [10/Jan/2024:12:05:15 +0000] "GET /product/123 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.12 - - [10/Jan/2024:12:10:25 +0000] "GET /home HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.13 - - [10/Jan/2024:12:15:40 +0000] "GET /cart HTTP/1.1" 302 "Mozilla/5.0"
192.168.1.14 - - [10/Jan/2024:12:20:00 +0000] "GET /checkout HTTP/1.1" 500 "Mozilla/5.0"
192.168.1.15 - - [10/Jan/2024:12:25:50 +0000] "GET /product/456 HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.20 - - [10/Jan/2024:12:50:10 +0000] "GET /cart HTTP/1.1" 200 "Mozilla/5.0"
192.168.1.21 - - [10/Jan/2024:12:55:55 +0000] "GET /checkout HTTP/1.1" 500 "Mozilla/5.0"
```

Your task is to use MapReduce to analyse the logs and extract the information - **the Top 10 most frequently accessed URLs within the hour peak hour (the hour with the most requests), and the number of unique visitors to each of the Top 10 URLs within the peak traffic hour**. You need to break down this complicated task into multiple MapReduce jobs and justify how the jobs can lead to the desired result in the end.

In your report, you are required to:

a) Describe the decomposition of the task into a series of MapReduce jobs and justify the reasoning. For each MapReduce job, identify the input, output, Mapper function, and Reducer function.

b) Write python programs for the map and reduce functions and run the MapReduce jobs in the **Hadoop** local setup. The code should be included **in an appendix** to the report and is excluded from the 4-page length limit.

c) **In the main report**, you need to provide screenshots for the generated result of each MapReduce job. Please note that the results should be **the execution results of Hadoop**, not the local testing results for your Python programs.

Task 3. Understanding Parallel Programming with Spark [35 marks]

Pi is an essential number in mathematics. It is both transcendental, which means it cannot be determined as a solution to an algebraic equation (polynomial with integer coefficients) and irrational, which means it cannot be expressed as a ratio of integers. Determining its value is not so simple. Here is a method which uses random numbers.

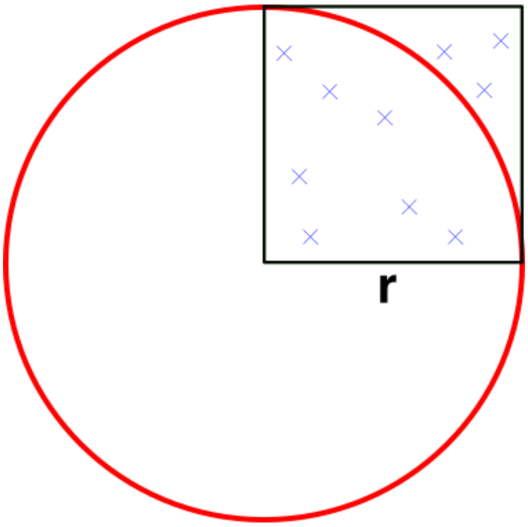

Figure 1: Unit circle with radius $r = 1$ and a square with side of length one.

Figure 1 is a unit circle, with radius $r = 1$ and a square with unit length side.

The area of the square is $A_s = r \times r = 1$ (1)

The area of the circle is $A_c = \pi r^2 = \pi$ (2)

The area of the upper quadrant contained in the black square is a quarter of the circle. Darts are thrown at random into the black square and counted to see how many fall within the quadrant. The number of darts inside the quadrant is proportional to the area, and so π can be estimated by taking the ratio of darts that fall inside and outside.

$$\frac{number\ of\ darts\ in\ quadrant}{number\ of\ darts\ in\ square} = \frac{\frac{1}{4}A_c}{A_s} = \frac{\pi}{4}$$

Here is the function in python using random numbers for x and y coordinates to compute whether a dart landing inside or outside of a unit circle.

```
import random

# function to compute whether a "dart" is inside the unit circle

def sample(p):

   x, y = random.random(),
   random.random()

   return 1 if x*x + y*y < 1
   else 0
```

Based on the key computation given above, please produce a python program to calculate Pi using **pyspark**. The program should be able to accept the number of samples (darts) and the number of partitions as two input arguments, and output the value of Pi given the input values.

```
from pyspark import SparkContext import sys


#Define the sample() function here


# You should start from creating a Spark Context
sc=SparkContext(appName="miPi")


# Take the number of samples
NUM_SAMPLES=int(sys.argv[1])


# Next you should parallelise the samples in Spark.
# Remember to take the number of partitions from an input argument


# Your code of using spark should go here.
#The number of darts inside the quadrant should be stored in an integer variable count



# Calculating Pi, python3 syntax
print("Pi:"+str(NUM_SAMPLES)+":"+str(4.0*count/NUM_SAMPLES))
```

You should also investigate how the accuracy of the calculation scales with the number of darts and the number of parallel elements.

In the report, you are required to include the following components:

a) Screenshot of your python code (**as an appendix**, excluded from the length limit)
b) **In the main report**, two plots respectively showing how the accuracy varies versus
   a. the number of darts at a fixed partition number;
   b. the number of partitions at a fixed number of samples.
c) Brief discussion and justification on the observations. Explore the reasons behind unexpected observations.

2. Assignment Submission Requirements

Front page of the submission

Module Code:

Assignment report Title:

Student Number (e.g. 25098635):

Date (when the work completed):

Actual hrs spent for the assignment:

Content of the required work

Please organise your report in terms of the coursework description headings.

A report in pdf format, up to 4 pages excluding the title page, diagrams, graphs, code and references.

Arial font 11pt or similar reasonable choice, but not smaller than 11pt.

4. Marking Scheme

Section ID	Content	Range for Marking
1.a.	Describing the semantic properties for the chosen distributed system	0 – 6
1.b.	Analysis of the software wish-list for the distributed system	0 - 14
1.c.	How does the CAP theorem apply to the distributed system?	0 - 3
	Why is the CAP theorem applied in such a way?	0 - 2
2.a	Decomposition of the task	0 - 10
	Justification of the decomposition	0 - 5
2.b	Python program for the MapReduce jobs	0 - 15
2.c	Results of the python program	0 - 5
3.a.	Pyspark code to compute Pi	0 - 12
3.b	Scaling results in plots – accuracy vs. number of darts	0 - 5
	Scaling results in plots – accuracy vs. number of partitions	0 - 5
3.c.	Discussion of the observations	0 - 6
	Justification of the observations including exploring reasons for unexpected results.	0 - 7
	Presentation, including report length, academic writing, report structure, figure format, reference style, typos.	0 - 5
	Total	0 - 100