NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF APPLIED SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

PARALLEL AND DISTRIBUTED COMPUTING

SCS4212

Examination Paper

This examination paper consists of 6 pages

Time allowed: 3 hours

Total Marks: 100

Special Requirements: None

Examiner: Mrs S Ndlovu

External Examiner: Dr C. Gombiro

INSTRUCTIONS TO THE CANDIDATES

1. Answer any four (4) questions.
2. Each question carries 25 marks.

MARK ALLOCATION

QUESTION	MARKS
1.	25
2.	25
3.	25
4.	25
5.	25

QUESTION ONE

a) Define parallel computing and distributed computing in your own words, showing how they differ in terms of task execution and resource management. [8]

b) Discuss the **three** key advantages and **three** disadvantages of parallel computing in the context of high-performance computing (HPC). [6]

c) Explain the importance of network latency in distributed computing. How can high latency impact the performance of distributed systems? [3, 3]

d) Provide, with justification, a real-world example where you would choose parallel computing over distributed computing. [5]

QUESTION TWO

You are building a distributed image processing system. A master process sends image processing tasks to worker processes. The workers process the images and send the results back to the master. See code snippets in Appendix I:

a) Explain in simple terms what Inter-Process Communication (IPC) is and why it is needed in this image processing system. Give **two** examples of how the master and worker processes communicate in this system (refer to the provided code examples if helpful). [3, 4]

b) The master process uses a queue called imageQueue to store the images waiting to be processed. Explain why a queue is a suitable data structure for this purpose. What are two benefits of using a queue in this scenario? Relate your answer to how the master and worker processes interact. [4, 4]

c) Imagine that the workers use Object Streams (ObjectOutputStream and ObjectInputStream) to send images and processing results. What is the purpose of these Object Streams? Why is it important to convert the image data into a stream of bytes before sending it over the network? [3, 2]

d) The master process uses multiple threads to communicate with different worker processes. Why is it helpful for the master to use multiple threads? What could happen if the master only used a single thread? Relate your answer to the concept of concurrency. [2, 3]

QUESTION THREE

A multinational cloud computing company is designing a hybrid computing system that integrates parallel and distributed architectures to handle high-performance computing (HPC) tasks and globally distributed applications. Using your knowledge of shared memory, synchronisation, consistency models, fault tolerance, and emerging trends, answer the following:

a) Explain the key differences between shared memory in parallel systems and distributed shared memory (DSM) in distributed systems. [5]

b) The company must ensure synchronisation between different processes in both parallel and distributed environments. What synchronisation challenges will they face, and what solutions can be used in each case? [2, 3]

c) Discuss how different memory consistency models (sequential consistency vs. relaxed consistency) impact performance and correctness in a hybrid system. [5]

d) Fault tolerance is a critical requirement for this system. Explain how checkpointing and data replication can help improve reliability in both parallel and distributed shared memory systems. [5]

e) Cloud computing introduces new trends in shared memory architectures. What emerging technologies (e.g., RDMA, chiplets, multi-core advancements) can be used to optimize memory performance in this hybrid system? [5]

QUESTION FOUR

Write a Java program that calculates the average of an array using threads. The array should store twenty (20) elements and the program should implement four (4) threads. Each thread should handle an equal part of the array. Use *Thread* class or *ExecutorService*.

Example Output:

Average computed by thread 1: 45
Average computed by thread 2: 22.5
:
Average computed by thread 5: 7.5
Final Average: 29.2

[25]

QUESTION FIVE

a) Using a diagram, describe the five key components of a remote procedure call (RPC system). [10]

b) Redraw the table below and outline the key differences between message passing in parallel and distributed systems. [15]

Feature	Message Passing in Parallel Systems	Message Passing in Distributed Systems
Definition		
Memory Model		
Communication Medium		
Latency & Bandwidth		
Synchronization		
Fault Tolerance		
Use Cases		

Figure 1: Key differences between message passing in parallel and distributed systems

APPENDIX I

1. Master process

```
1  import java.net.*;
2  import java.io.*;
3  import java.util.concurrent.*;
4  public class Master {
5  private static final int PORT = 8080;
6  private static final int NUM_WORKERS = 4; // Number of worker processes
7  public static void main(String[] args) throws IOException, ClassNotFoundException,
8  InterruptedException {
9  ServerSocket serverSocket = new ServerSocket(PORT);
10 ExecutorService workerPool = Executors.newFixedThreadPool(NUM_WORKERS);
11 // Thread pool for workers
12 BlockingQueue&lt;Image&gt; imageQueue = new LinkedBlockingQueue&lt;&gt;(); //
       Queue
13 for images to process
14 BlockingQueue&lt;Image&gt; resultQueue = new LinkedBlockingQueue&lt;&gt;(); //
       Queue
15 for processed images
16 // Start worker processes (in a real scenario, these would be separate programs)
17 for (int i = 0; i &lt; NUM_WORKERS; i++) {
18 new Thread(() -&gt; {
19 try {
20 Socket workerSocket = new Socket(&quot;localhost&quot;, PORT + 1 + i); //
       Different
21 ports for workers
22 processImages(workerSocket, imageQueue, resultQueue);
23 } catch (IOException | ClassNotFoundException | InterruptedException e) {
24 e.printStackTrace();
25 }
26 }).start();
27 }
28 // Accept client connections and add images to the queue
29 while (true) {
30 Socket clientSocket = serverSocket.accept();
31 try (ObjectInputStream in = new
32 ObjectInputStream(clientSocket.getInputStream())) {
33
34 Image image = (Image) in.readObject();
35 imageQueue.put(image); // Add image to processing queue
36 }
37 }
38 }
```

```
39- private static void processImages(Socket workerSocket, BlockingQueue&lt;Image&gt;
40  imageQueue, BlockingQueue&lt;Image&gt; resultQueue) throws IOException,
41- ClassNotFoundException, InterruptedException {
42- try (
43  ObjectOutputStream out = new
44  ObjectOutputStream(workerSocket.getOutputStream());
45  ObjectInputStream in = new ObjectInputStream(workerSocket.getInputStream())
46- ) {
47- while (true) {
48  Image image = imageQueue.take(); // Take image from queue (blocking)
49  out.writeObject(image); // Send image to worker (blocking)
50  Image processedImage = (Image) in.readObject(); // Receive processed image
51  (blocking)
52  resultQueue.put(processedImage); // Add processed image to results queue
53  }
54  }
55  }
56  }
```

2. Workers process

```
1- import java.net.*;
2  import java.io.*;
3- public class Worker {
4- public static void main(String[] args) throws IOException,
       ClassNotFoundException {
5  int port = Integer.parseInt(args[0]); // Get port from command line arguments
6  Socket socket = new Socket(&quot;localhost&quot;, port);
7- try (
8  ObjectInputStream in = new ObjectInputStream(socket.getInputStream());
9  ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream())
10- ) {
11- while (true) {
12  Image image = (Image) in.readObject(); // Receive image from master
13  (blocking)
14  Image processedImage = processImage(image); // Process the image
15  out.writeObject(processedImage); // Send processed image back to master
16  (blocking)
17  }
18  }
19  }
20
21- private static Image processImage(Image image) {
22  // Image processing logic here (e.g., resize, filters)
23  // ... (This is where the image manipulation happens) ...
24  return image; // Return the processed image (in this example, the same image)
25  }
```

END OF QUESTION PAPER