Indian Institute of Information Technology Chittoor, Sri City

Name of the Exam: **Distributed Computing** Duration: **1½** Hours

Total Marks: 50 / 50

Roll No.:	I	S	2	0	1	0	1	1	2	7	

Name : Alfred De Souza

Room No.: 314

Invigilator's Signature: ________________________

Seat No. : A7

Instructions:

1. Read all questions carefully and answer them in the space provided (**Strictly within the box**).
2. Answer all questions compulsorily (no choice, unless otherwise mentioned) within the given box. **Answers must be written inside the box only** and no rough work should be done inside the box. Use your space insider the answer box efficiently and avoid useless explanations.
3. Two empty sheets are provided at the end of this booklet for rough work. No additional sheet would be provided.
4. **Calculators / Electronic gadgets** are **NOT** permitted during the examination.
5. Most importantly, **NO answer should be written in Pencil**. Pencils are allowed for rough work but the final answers should be written using either a **BALL - POINT** pen or **INK** pen.
6. We already provided two empty sheets for rough work. Use pencil to do the rough work so that you could reuse these extra pages.
7. Exchanging of stationary items is prohibited inside the hall. Bring your own stationary items.

---

1. [4 Marks] a) Define a reachable state. b) When did a distributed algorithm say to be terminated? (State two conditions). c) When does a termination detection algorithm restart?

a) Define a Reachable State:

A state S' is reachable from a state S if there exists a consistent run (Ordering of events that satisfies all the happened-before relations) from S to S'

b) Termination Conditions: A distributed computation is said to be terminated

1) if and only if all processes have become idle

2) there should be no message in transit in any channel

c) Restart Termination Detection:
(1) Whenever processes are not idle
(ii) There exists at least one message in transit

2. [5 marks] Consider the following parameters:
   Let r be the replication factor, s be the initial size of data, i be the intermediate data factor and d be the disk space available per node.
   Now using the following values: $r = 4$, $s = 10.684$ TB, $i = 33.367\%$, $d = 2$ TB
   Calculate
   (a) HDFS nodes storage (with the assumption that no compression is used)
   (b) The number of data nodes
   (c) The cluster size irrespective of all name nodes (primary or secondary)

```
(a) HDFS Nodes Storage:
    Calculate Hadoop Storage(H) = c * r * S / (1-i)
    = 1 * 4 * 10.684 TB / (1-33.367%)
    = 42.736 TB / 0.66633
    = 64.1364 TB

(b) Number of Nodes Required:
      Consider, each data node is 2TB
      Number of datanodes (D) = Hadoop Storage / Disk space per node
                              = H / d = 64.1364 TB / 2 TB = 32.0682 = 33

(c) Cluster Size:
      If we need 1 namenode and 2 secondary namenodes, that is, Master(M) = 3
      Cluster size = M + D = 33 + 3 = 36, So, we need a cluster size of 36 nodes
```

3. [4 marks] State any 4 advantages of the Hadoop Framework with a brief description in 1-2 lines:

```
Advantages of the Hadoop Framework:
  (1) Scalable: Store and distribute very large data sets across hundreds of inexpensive
      servers that operate in parallel ...  run applications on thousands of nodes
      involving thousands of terabytes of data

  (2) Cost-Effective: Instead of spending a huge money, hadoop offers computing and
      storage capabilities at a cheaper price per terabyte

  (3) Fast: Tools are on the same machine where data is present. This results in much
      faster data processing

  (4) Fault-Tolerant: Data is replicated across nodes in a cluster. So in the event of
      one node failure, same chunk of data can be obtained from another machine
```

4. [5 Marks] Describe the formal description of the Termination Detection algorithm that uses distributed snapshots. Also mention the best, average and the worst-case message complexity.

Description (of TD using Distributed Snapshots):

Marker Sending Rule for process i:
  Process i records its state
  For each outgoing channel C on which a marker has not been sent, process i sends a marker along C before Process i sends further messages along C

Marker Receiving Rule for process j:

On receiving a marker along channel C:
if process j has not recorded its state then
  Record the state of C as the empty set and follow the "Marker Sending Rule"
else
  Record the state of C as the set of messages received along C after process j's state was recorded and before process j received the marker along C

Message Complexity: For single instance of the algorithm (n = number of edges)

Best Case	Average Case	Worst Case
O(n) messages	O(n) messages	O(n) messages

5. [5 Marks] Describe LCR algorithm for Leader Election and perform the following analysis: correctness, time and message complexity in the worst-case scenarios.

Description:

Network: Directed Ring Network with n processes and each process has a unique ID.
Steps:

- Each process sends its ID around the ring
- When a process receives a ID, it compares this one to its own
  - If the incoming ID is greater, then it passes this ID to the next process
  - If the incoming ID is smaller, then it discards it
  - If it is equal, then the process declares itself the leader

Analysis:

(a) **Correctness:**
Elects the process with the largest ID.
Message containing largest id passes through every processor

(b) **Time Complexity:**
$O(n)$ Computation time

(c) **Message Complexity:**
$O(n^2)$ messages

6. [3 Marks] The LCR algorithm is of O(n^2) complexity for Leader Election in a ring network that consists of n nodes. Deduce it to an algorithm with O($n \log n$) message complexity.

Deduce O(n^2) → O($n \log n$) :

Idea: Let us allow messages containing smaller ids travel smaller distance in the ring
Steps:
- Each proc. tries to probe successively larger neighborhoods in both directions
  - size of neighborhood doubles in each phase
- If probe reaches a node with a larger id, the probe stops
- If probe reaches end of its neighborhood, then a reply is sent back to initiator
- If initiator gets back replies from both directions, then go to next phase
- If proc. receives a probe with its own id, it elects itself

7. [4 Marks] List out and explain various types of topology abstractions in 1-2 lines. Which one is more efficient in terms of a specific task given in hand?

Types of Topology:

Physical topology: All LAN, WAN links, and direct edges between end hosts

Logical topology: Nodes are end hosts where application executes and edges are logical channels among these nodes. Ex., Fully connected or any subgraph – partial system view, needs multi-hop paths

Superimposed topology (also called as "topology overlay"): Selected portion of a logical topology based on specific criteria for in-depth tasks. Examples: ring, tree, mesh, hypercube

Efficient Abstraction:

Topology overlays – These overlays are dynamic in nature and can be explored for efficient information gathering and analysis tasks.

8. [3 Marks] Draw the Hierarchy of the Models of Communication pictorially.

Models of Communication:

- Models of Communication
  - 1-to-1
    - Unicast
      - 1 – 1
      - point-to-point
    - Anycast
      - 1-to-nearest 1
  - 1-to-many
    - multicast
      - 1-to-many
      - Group Comm
    - broadcast
      - 1-all

9. [4 Marks] Briefly describe the possible failures that could take place in a distributed system.

Failures:

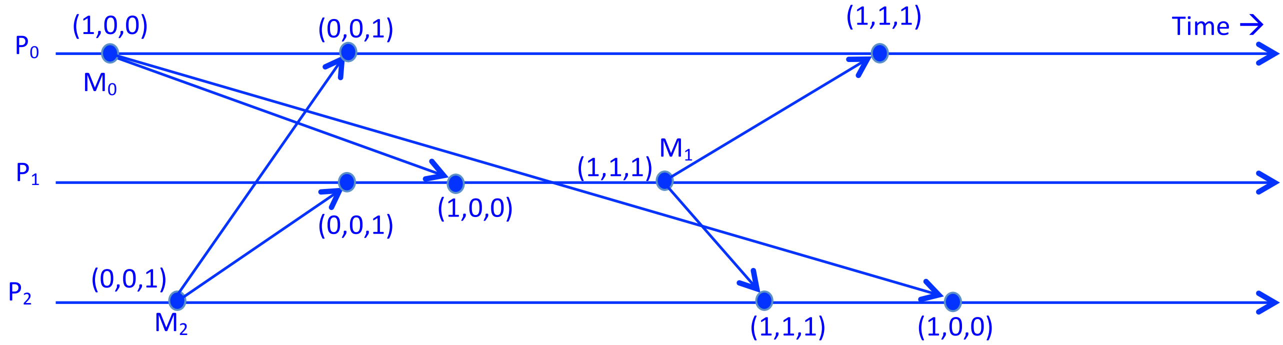
Crash failure:
Process stops communicating

Omission failure (typically due to network): There are two types of omissions.
Send omission: A process fails to send messages
Receive omission: A process fails to receive messages

Byzantine Failure:
Some messages are faulty, including sending fake messages

Partition Failure:
The network may get segmented, dividing the group into two or more unreachable sub-groups

10. [4 Marks] Carefully look at the state – time diagram of 3 processes in a distributed system. Assume that the initial vector at each P_i for all i, is (0,0,0). A group of 3 processes sends out multicast messages using causal ordering. Now illustrate a situation in which the message is kept in the hold-back queue and eventually how is the message delivered?



Analysis:

At P_2, V_1 = (1, 1, 1) and V_2 = (0, 0, 1). When M_1 is sent from P_1 to P_2, it is put in hold-back queue. Since (i) $V_1[1] = V_2[1] + 1$; but (ii) $V_2[0] < V_1[0]$. This means that there was a message receipt at P_1 from P_0 before the sending of M_1 from P_1

Now M_0 from P_0 is delivered at P_2. At P_2, V_0 = (1, 0, 0) and V_2 = (0, 0, 1). Now (i) $V_0[0] = V_2[0] + 1$; and (ii) $V_0[1] < V_2[1]$ and, $V_0[2] < V_2[2]$; This implies that M0 is delivered. Now check with M_1 that was put in hold-back queue. Now M_1 will follow the above axioms and eventually the message M_1 is delivered at P_2

11. [3 Marks] Consider the multi-cast algorithm that uses causal ordering. How does the causal ordering algorithm handle send and receive messages at a specific process?

Causal Ordering Algorithm:

- When P_j sends a message, it increments its own entry and sends the vector
  - $V_j[j] = V_j[j] + 1$
  - Send V_j with the message
- When P_i receives a message from P_j
  - Check that the message arrived in FIFO order from P_j: $V_j[j] == V_i[j] + 1$?
  - Check that the message does not causally depend on something P_i has not seen
    $\forall k,\ k \neq j: V_j[k] \leq V_i[k]$?
  - If both conditions are satisfied, P_i will deliver the message
  - Otherwise, hold the message until the conditions are satisfied

12. [2 Marks] Define "Synchronization Delay" in distributed systems.

Synchronization Delay:

Synchronization Delay is the time difference between a process leaving Critical Section and a process entering into the Critical Section next time.

It is desirable to minimize the synchronization delay in distributed systems.

13. [4 marks] Describe the steps in the Lamport's algorithm for distributed mutual exclusion. Prove that this algorithm guarantees mutual exclusion, avoids starvation and deadlock.

Algorithm:

Requesting CS:

Send REQUEST(ts_i, i) where (ts_i,i) is the request timestamp;
Place REQUEST in request_queue$_i$

On receiving the message, P_j sends time-stamped REPLY message to P_i;
P_i's request is placed in request_queue$_j$

Executing CS:

P_i has received a message with time stamp larger than (ts_i, i) from all other sites.
P_i's request is the top most one in request_queue$_i$

Releasing CS:

Exiting CS: send a time stamped RELEASE message to all sites in its request set.
Receiving RELEASE message: P_j removes P_i's request from its queue

Analysis:

- Purpose of REPLY messages from P_i to P_j is to ensure that P_j knows of all requests of P_i prior to sending the REPLY
- Synchronization delay = max message transmission time
- Requires FIFO channels and requests are granted in order of increasing timestamps
- 3(N-1) messages per CS: (N - 1) REQUESTs; (N - 1) REPLIES; (N - 1) RELEASE messages

Space for ROUGH WORK

Space for ROUGH WORK