Name - Vishal Yadav
Roll No - 2301010258

Assignment - 04

OPERATING SYSTEM

Que 6 Distributed Deadlock detection

Given

S_1 $P_1 \rightarrow P_2$, $P_3 \rightarrow P_4$

S_2 $P_2 \rightarrow P_5$, $P_5 \rightarrow P_6$

S_3 $P_6 \rightarrow P_1$

a combine fragments into the global wait for graph

From all sites, the combined edges are -

$P_1 \rightarrow P_2$, $P_3 \rightarrow P_4$, $P_2 \rightarrow P_5$, $P_5 \rightarrow P_6$, $P_6 \rightarrow P_1$

So, Global wait for graph is

$P_1 \rightarrow P_2 \rightarrow P_5 \rightarrow P_6 \rightarrow P_1$ and $P_3 \rightarrow P_4$

b Detect if a deadlock exists

Yes

Cycle ~~detected~~ detected $P_1 \rightarrow P_2 \rightarrow P_5 \rightarrow P_6 \rightarrow P_1$

Thus, processes P_1, P_2, P_5, & P_6 are involved

c Distributed algorithm

use, the Chandy - Misra Haas (CMH) algorithm for distributed - deadlock detection

- Each site sends probe message (initiator, Sender receiver)

- If a probe returns to the initiator, a deadlock is detected

- this algorithm works efficiently without requiring a global co-ordinator

(7) Distributed file system

Given -
- local access time = 5ms
- Remote access time = 25ms
- Probability of being remote = 0.3

(a) expected file access time
$E(T) = (1-0.3)(5) + (0.3)(25)$
$= 0.7(5) + 0.3(25)$
$= 3.5 + 7.5 \Rightarrow 11\ ms$

b) Caching strategy - use client side caching with least Recently used (LRU) Replacement
- Frequent accessed remote files are cached locally
- Reduce remote access frequency, lowering avg access - time
- LRU ensures cache freshness and avoid stale data build up

returns to the initiator, a deadlo

efficiently without requiring a

(8) Given -
Full checkpoint = 200 ms
Incremental checkpoint = 50 ms
Must recover within 1 sec (1000 ms RPO)
Total operation window = 10 sec

a) optimal mix
To minimize overhead while ensuring quick recovery

- Take 1 full checkpoint every 10 sec
- Incremental checkpoints every 2 sec

That Gives

- 1 full checkpoint = 200 ms
- 4 incremental checkpoints = 4 x 50 = 200 ms
- Total overhead = 400 ms in 10 s (4 %)

b) Explanation

Incremental checkpoints save only changed state reducing data volume

combining full - incremental ensures faster recovery (restore last full - few incremental)

(9) a) Scheduling challenges & load balancing algorithm

Challenges -
- uneven global traffic (eg flash sales in one region)
- latency due to geographical distribution
- Dynamic scaling under sudden demand

b) Fault tolerance strategy - use Active - Active Replication across region with:

- Data replication (synchronous/asynchronous) for redundacy
- Failover mechanism via global DNS routing
- Regular checkpoints

Ans 1

A race condition occurs when two or more entities try to change a shared resources simultaneously, leading to unpredictable results

eg. (real world) - Two people editing the same document at once - one saves changes while the other overwrite them

Ans- Drawback - Process must request all resource at once before execution begins, leading to resources underutilization & reduced concurrency since some resources remain idle for long periods