Operating System Assignment - 4

Rohini
2301010266

Date
Page

1. Distributed Deadlocks Detection Simulation –

→ Fragments :

- S_1 : $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4$
- S_2 : $P_2 \rightarrow P_5$, $P_5 \rightarrow P_6$
- S_3 : $P_6 \rightarrow P_1$

a) Global wait-for Graph (combined)
$P_1 \rightarrow P_2 \rightarrow P_5 \rightarrow P_6 \rightarrow P_1$ (cycle) Also $P_3 \rightarrow P_4$ (separat

b) Deadlock ?
→ Yes Process Involved in dead lock cycle
P_1, P_2, P_5, P_6.

c) Suggested distributed algorithm
Use the chandy - Misra - Hass edge-chasing (Probe) algorithm for distributed deadlock detection each site send Probes, along wait-for edges to detect cycles without centralized Graph assembly.

2. Distributed File System Performance :
Given : local = 5ms, remote = 25 ms
Prob (remote) = 0.3

a) Excepted access time :
$E = 0.7 \times 5ms + 0.3 \times 25\ ms = 35 + 37.5 = 11ms.$

b) The chaching Strategy:
Client - Side Read cache with LRU + TTL - based validation

→ Justification. Frequently-Read Remark files will be served locally reducing Remote access (0.3 fraction). LRU evicts less-used items TTL keeps staleness bounded. Implanes average latency while keeping consistency manageable.

3. Checkpointing Mix to Meet RPO < 1s

Sol Given,

Full = 200ms, incremental = 50ms, RPO = 1s

a) Proposed mix (over 10s)

- Take one full checkpoint every 10s (at t = 0 in the period)
- Take incremental checkpoint every 2s (at t = 4, 2)

Total overhead (Per loss) : 1 x 200ms + 9 x 50 ms = 200 + 450 = 650ms.

b) Reasoning.

- With incremental every 1s, the maximum work lost on failure < 1s → meets → RPO
- Full once per 10s bound Recovery time 150 restricting an older full + a small no of Incrementals is feasible)
- This mix minimize full checkpoint cost while keeping incrementals frequently high enough to meet RPO.

4. Case Study - Global E-commerce Platform.

a) Distributed scheduling challenges FRO flash sales.

- Massive Sudden spikes in Request.

Date
Page

- Geographic distribution & latency data locality matters for latency & inventory correctness
- Heterogeneous nodes

b) Fault - tolerance Strategy (RTO & RPO):

- Active - active multiple - region development:

sol Service Run concurrently in multiple Regions so failure is seamless.
(RTO ≃ near-zero at service lead)

- Data Strategy:

→ Critical transactional data: Use synchronous replication within the region to guarantee consistency & low RPO; cross Region replication but with frequent replication to keep RPO small

→ Data log / less-critical data: Uses multi-region eventual consistency with frequent async replication & local caches.

- Recovery Plans: Automated failure by health check & Global load balance, warm stand by replices ready to accept traffic, & durable messaging to buffer incoming requests during failure
- Operational measures: chaos testing, backups, runbook & automated scaling to minimize RTO, use snapshots + incremental backups to band RPO

Satish 21/11/25