

Assignment 04

Ques 1. To form the global wait-for graph (combined) :-
given :-

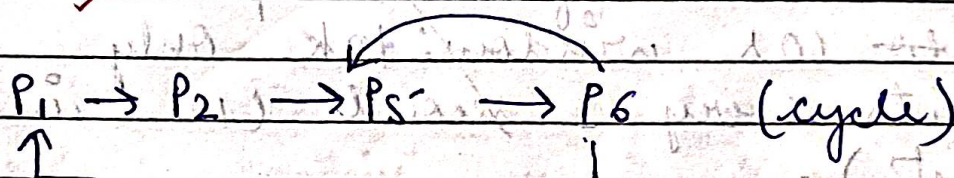
from all 3 sites

S₁ P₁ → P₂, P₃ → P₄

S₂ P₂ → P₅, P₅ → P₆

S₃ P₆ → P₁

graph :-



P₃ → P₄ (not part of cycle)

(b) There is a cycle.
P₁ → P₂ → P₅ → P₆ → P₁

Hence, deadlock exists and processes involved P₁, P₂, P₅, P₆. Processes P₃ & P₄ are not part of the deadlock.

(c) Chandy-Mishra has probe-based deadlock detection algorithm :-

→ Each site sends a probe message along wait-for edges.

→ If a probe returns to its origin, a cycle is detected.

→ Works well in distributed environments where no single site has the entire graph

ques2 Expected access time = $0.3 \times 25 \text{ ms} + 0.7 \times 5 \text{ ms}$
 $\Rightarrow 7.5 + 3.5$

One catching strategy to improve performance is popular files are served locally. Leases keep caches reasonably consistent without excessive invalidation traffic giving a good tradeoff between latency and correctness.

ques3 (a) Recommended Schedule (Over 10s):
 \rightarrow If a recent full checkpoint exists before the 10s window: take only incremental checkpoints every 1 second (10 incremental checkpoints).

\rightarrow If no prior full exists: take one full checkpoint at the start, then incremental checkpoints every 1 second (1 full + 10 increments across the period).

overhead calc: -
with prior full = $10 \times 50 = 500 \text{ ms}$
without prior full = $200 + 10 \times 50 = 700$

b) RPO \rightarrow 1s means you must capture changes at least every second. Incremental checkpoints are much cheaper (50 ms) so scheduling them every second minimizes overhead while meeting RPO. A full checkpoint is only needed occasionally to bound restore complexity - otherwise pure incrementals plus an earlier full give the lowest cost.

Ques 4* Scheduling Challenges are :-

sudden traffic spikes, hot product partitions, cross-region latency, uneven load, data locality vs consistency tradeoffs.

Suggested algorithm uses power-of-d-choice (P2C) front-end routing combined with Rendezvous / consistent hashing for session affinity & data locality, P2C gives very low queuing imbalance, consistent hashing routes requests user data to reduce latency.

* Fault tolerance strategy includes :-

- deploy active-active multi-region services
- critical metadata / state
- Bulk data / catalogs
- using periodic full + incremental backups
- For RTO :-

automate failover and warm standbys so services in another failover region take traffic in seconds.

For RPO :-

ensure critical data is synchronously replicated or use cross-region commit protocols.

Strides

21/11/25