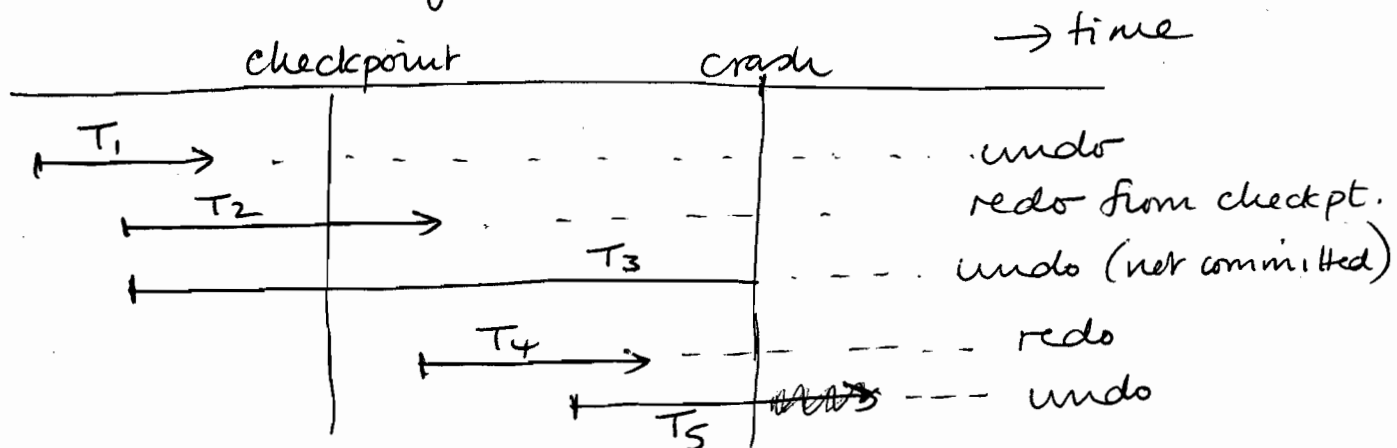


999 Concurrent Systems solution a.

Paper 4 q. 2

a) Loss of main memory in a centralised T.P. system. JMB

1. Assume we update database records in place but first must write a log record containing e.g. before & after value ^(discuss).
2. Assume a checkpoint is written to disc periodically (containing) all current log records, all current database records & a record of transactions in progress. Tx's may be in the following states:-



To carry out the actions we can start from the checkpoint with

~~UNDO~~-list = T₂, T₃.

process the log to the end ~~in redo~~ adding any new Tx to the ~~UNDO~~ list & moving any committed tx to the REDO-list. At the end:-

UNDO-list = ~~T₂~~ T₃ ~~T₄~~ T₅

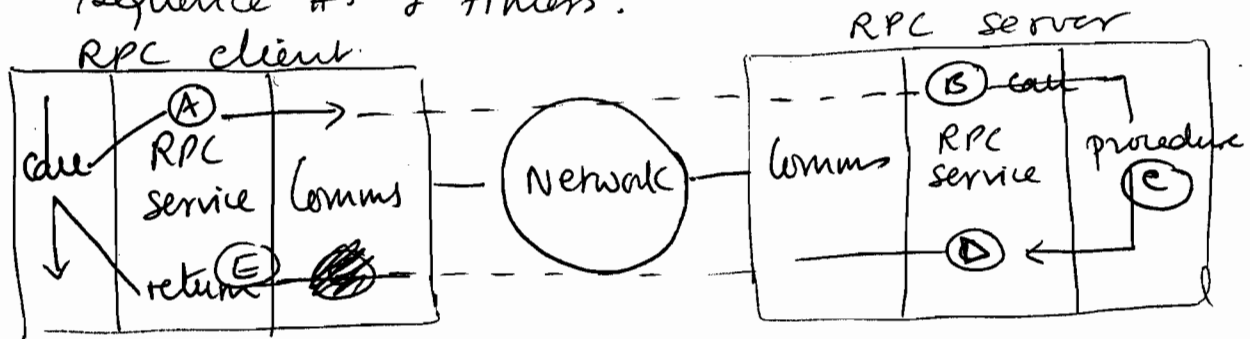
REDO-list = T₂ T₄

Work back through the log from the end to the checkpoint (and before to the start of relevant txs)

undoing (in this case) T₃ and T₅. Then redo the operations of T₂ and T₄ in the database and log

Note there can be a crash at any time including during checkpointing and during the restart procedure. Log records must make idempotent undo/redo possible.

- b) RPC protocol - need to describe it including sequence #'s & timers.



RRAP protocol:

- A assign sequence #
marshal argument
set timer (default or specific)
pass to lower level (eg on a socket).
- B note id
unmarshal argument
make local call
- C do procedure & return to RPC service
- D marshal reply params.
set timer
pass to lower level
- E receive - note id - unset timer at A.
send ACK (D unsets timer)
unmarshal argument
return to caller.

Loss of memory at client after RPC sent

timer at D expires - retry - ~~hard error exception~~ ^{discard} to caller

higher levels must undo any state change at C
don't exterminate program at this level - no RPC problem

Loss of memory at server while RPC in progress -

timer at A expires - retry - hard error exception to client
if crash after state change - high level must undo.
no RPC service