

Concurrent Systems and Applications 2004

Paper 6 Question 4 (TLH)

This question is examining the material in the lectures on ‘Transactions, isolation, serializability’ and ‘Crash recovery and logging’.

- (a) A transaction processing system is using a *write-ahead log* as a mechanism for providing persistent storage. What information must be written to the log

- (i) when a transaction starts,

A transaction start record is recorded holding a unique identifier for the transaction.

- (ii) when a transaction performs an operation on a persistent object,

An update record is recorded holding the old and new state of the object.

- (iii) when a transaction commits? [2 marks each]

A commit record is recorded indicating the transaction involved. It must be written to persistent storage before the commit is deemed to have succeeded.

- (b) Describe how the log can be used to roll-back a transaction that has aborted or become deadlocked. [6 marks]

Proceed backwards through the log to find the transaction’s start record. Proceed forwards from that point examining the update records: restore the old value for updates made by this transaction. If an operation performed by this transaction conflicts with a subsequent one by another transaction then that transaction must also be aborted (using the same procedure, and potentially cascading again).

- (c) The log can also be used to recover after some kinds of system failure.

- (i) Describe how the log can be used to recover after a *fail-stop* crash. [2 marks]

In this basic scheme the log holds all of the updates ever made to the system. Maintain an *undo set* of transactions that have started but not committed: the set is initially empty, transactions are added to it when they start and removed from it when they commit. Replay log entries forwards, effectively re-running the entire execution. When the replay is complete, proceed backwards through the log and restore the old value in update entries for transactions in the undo set.

- (ii) What is meant by *checkpointing*? How will using it affect the structure of the log and the recovery procedure after a crash? [6 marks]

A checkpoint represents a ‘snapshot’ of the system. It is constructed so that recovery can proceed from the checkpoint rather than from the very start of the log. The checkpoints adds a new form of log entry which lists the set of

transactions active at the time of the checkpoint. A separate file indicates the most recent checkpoint.

To recover from a crash we must first build up a *redo set* (of transactions whose log entries must be replayed to the ‘new’ value) and an *undo set* (of transactions whose log entries must be replayed, as before, to the ‘old’ value). Initially the transactions active at the checkpoint are in the undo set and the redo set is empty. Proceed through the log from the checkpoint. Add transactions to the undo set if they start. Move transactions to the redo set if they commit.