Centralized Recovery

Submitted by Ravi Prakash Giri (rgiri8)

The chapter enlightens the reader with the concept of recovery algorithms for centralized systems. It starts off by highlighting various ways a system can fails and characterized the database system failures under three parts: transaction failures, system failures and media failures. The author has emphasized on `undo-redo logging` as it demonstrates most of the recovery problems and all technique must handle it. The chapter further expatiate conceptually on deferred updating, shadowing, checkpointing, and archiving.

The chapter is successful in providing a thorough details of the architecture of data manager including Stable Storage and the Cache manager. The `undo-redo rule` provided ensures that the last committed value of data item is available in stable storage without being dependent on each other(undo-redo). There are few points that this chapter fails to explain. The recovery techniques provided for system failure are only valid if the failure is detected. Although it's a valid point in case of transaction failure as transaction failure executes an `Abort` operation but it is possible that volatile or stable storage gets corrupted without failure being detected. Also, the assumption that the execution will be serialized and atomic for the Recovery Manager's five procedures may result in `cascading aborts`. Which means aborting one transaction may result in a need to abort a second transaction, and then a third, and so on. So, this will result in a waste of already partially executed transactions.

I think even the recovery should be resilient to failure too. So, if there is any failure during the recovery process and we start recovery again then the recovered state should be no different than if the second failure had not taken place. To ensure this, the stable DB should not be modified by the restart procedure in any manner that may affect the subsequent recovery.