Universität Freiburg
Institut für Informatik
Prof. Dr. P. Fischer/Prof. Dr. F. Kuhn

Georges-Köhler Allee, Geb. 51
D-79110 Freiburg
peter.fischer@informatik.uni-freiburg.de

Exercises
Distributed Systemes: Part 2
Summer Term 2015
30.6.2015
Solution Proposal

4. Exercise sheet: Recovery and Commit Coordination

Exercise 1
Judge whether the following statements about recovery are true or false. In your assessment, give clear reasons:

A. Recovery techniques are employed by DBMSs to guarantee the isolation and durability transactional properties.
B. The write-ahead logging principle (WAL) states that all log records for a given updated page should be written to stable storage before the page itself is written and that a transaction is only considered to be committed after all its log records have been written to stable storage.
B. Using an operating system or disks with an active write(-back) cache is safe for both log and data storage
D. When a no-force and no-steal strategy is used, one must implement schemes both for the redo of operations from commited transactions whose changes have not yet been written to the database and for the undo of operations from uncommited transactions that have already updated the database.
E. Physical logging implies that only actions, not images, processed by the system are written to the log.

Solution:

(a) False, isolation is not guaranteed by recovery (atomicity and durability!)
(b) True
(c) False, enabled write cache for log storage does not guarantee that the entry was written before the actual data is written. No atomic state: possible loss of data Possible for data disks, due to logs.
(d) False, correct for no-force and steal.
(e) False, correct for logical logging.

Exercise 2
Characterize centralized 2PC and linear 2PC with respect to

(1) message and time complexity,
(2) possibilities of processes to become uncertain.

Solution:

(a) Centralized 2PC requires 3n messages (if ACKs are not considered), and takes 3 rounds. Linear 2PC requires 2n messages, but needs also 2n rounds.
(b) For centralized 2PC, all participants are uncertain (in the same way) from the moment they cast their vote until they receive the decision from the coordinator. In linear 2PC, period of uncertaintly shrinks with the distance from the coordinator, the rightmost participant is actually never uncertain, whereas the leftmost participant is uncertain from 2n-2 rounds.

Exercise 3

Consider the following scenario. The server at a travel agency initiated a distributed transaction involving four other participants – an international airline company (P1), a domestic airlines (P2), a hotel chain (P3) and a car rental agency (P4). The travel agency acts as the coordinator (C) in this transaction and can communicate directly with all other participants. Assume that all parties agree to the transaction and are willing to commit.

A. Assuming that there are no failures, describe the sequence of messages exchanged and the log entries written at each site when using the 2PC protocol to carry out this transaction.

B. Now, suppose that participant P2 crashes just after sending the "vote-commit" message (i.e. this message is successfully received by the coordinator) and there are no further failures. Repeat the previous question for this modified scenario and show the final state of each participant including the coordinator. Are any of the participants blocked?

C. If you use the 3PC protocol instead of 2PC for the scenario in part [B] above, what would be the sequence of messages and log entries? What would be the final state of each participant?

Solution:

(a)
- a) Coordinator (C) writes log "Begin Commit"
- b) C sends "Prepare" msg to participants
- c) Each participant (P1-P4) writes log "Ready"
- d) P1-P4 sends "Vote Commit" msg
- e) C writes log "Commit"
- f) C sends "Global Commit" msg to all
- g) P1-P4 write log "Commit"
- h) P1-P4 send "Ack" msg
- i) C writes log "end_of_transaction"

(b)
- a) Coordinator (C) writes log "Begin Commit"
- b) C sends "Prepare" msg to participants
- c) Each participant (P1-P4) writes log "Ready"
- d) P1-P4 sends "Vote Commit" msg
- e) **P2 Crashes**
- f) C writes log "Commit"
- g) C sends "Global Commit" msg to all
- h) P1,P3,P4 write log "Commit"
- i) P1,P3,P4 sends "Ack" msg
- j) **C waits for "Ack" from P2 and continuously sends "Global-commit" to it after timeouts**

The coordinator is blocked; P1,P2,P4 are not blocked

(c)
- a) Coordinator (C) writes log "Begin Commit"
- b) C sends "Prepare" msg to participants
- c) Each participant (P1-P4) writes log "Ready"
- d) P1-P4 sends "Vote Commit" msg
- e) C writes log"Prepare-to-Commit"
- f) C sends"Prepare-to-Commit" msg
- g) **P2 Crashes**
- h) P1,P3,P4 write log"Prepare-to-Commit"
- i) P1,P3,P4 send"Ready-to-Commit" msg
- j) C timeouts in PRE-COMMIT state
- k) C writes "Commit" record in log
- l) C sends "Global-Commit" msg to all
- m) P1,P3,P4 writes log"Commit"
- n) P1,P3,P4 sends"Ack" msg and terminate

All except P2 have terminated, none is blocked