Co-funded by the
Erasmus+ Programme
of the European Union

Distributed Database Systems and Data Warehouses

Dr. Volodymyr Sokol
(vlad.sokol@gmail.com)

LECTION 6

Three-Phase Commit (3PC)

- 2PC *is not* a non-blocking protocol.
- For example, a process that times out after voting commit, but before receiving global instruction, is blocked if it can communicate only with sites that do not know global decision.
- Probability of blocking occurring in practice is sufficiently rare that most existing systems use 2PC.

Three-Phase Commit (3PC)

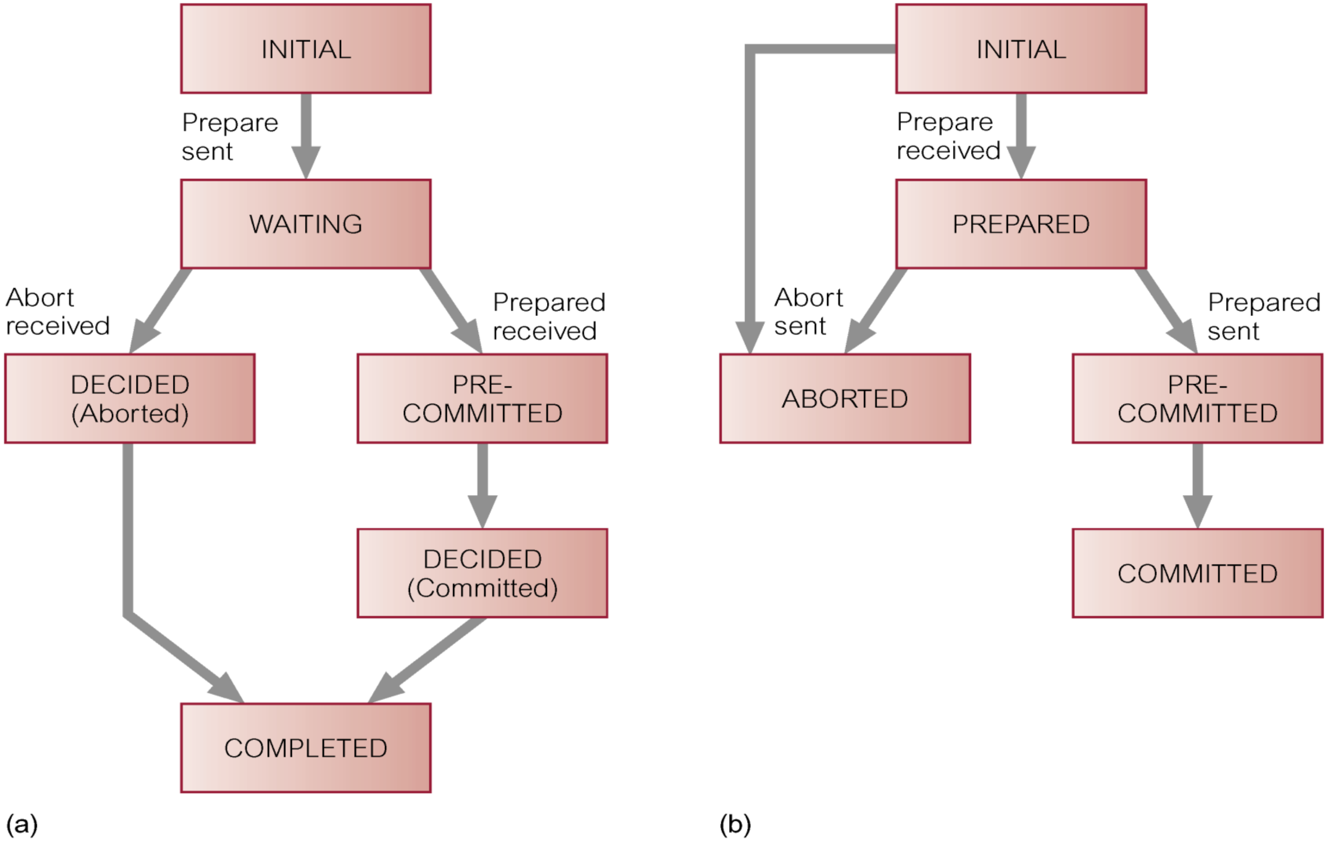
- Alternative non-blocking protocol, called *three-phase commit (3PC)* protocol.
- Non-blocking for site failures, except in event of failure of all sites.
- Communication failures can result in different sites reaching different decisions, thereby violating atomicity of global transactions.
- 3PC removes uncertainty period for participants who have voted commit and await global decision.

Three-Phase Commit (3PC)

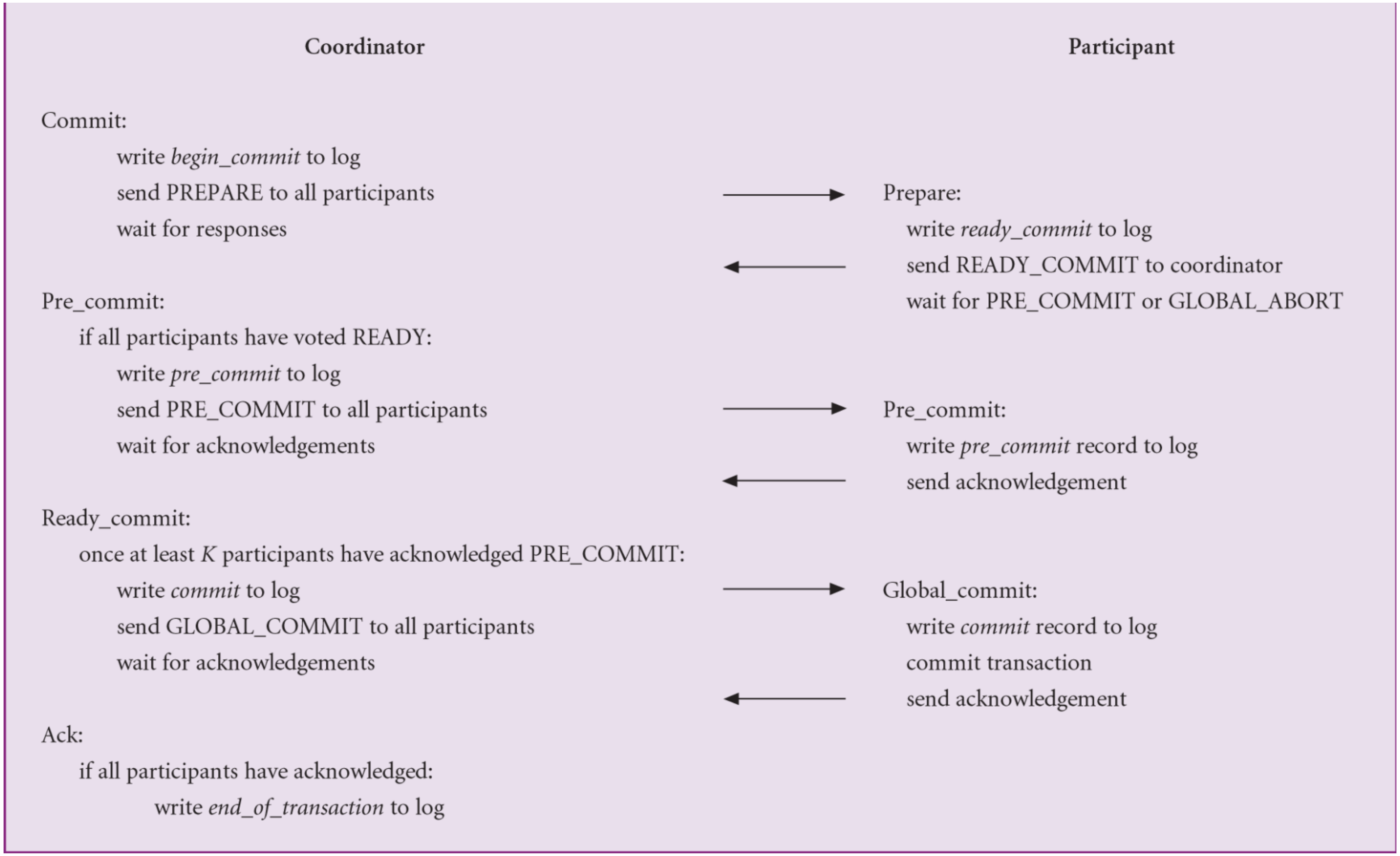
- Introduces third phase, called *pre-commit*, between voting and global decision.
- On receiving all votes from participants, coordinator sends global pre-commit message.
- Participant who receives global pre-commit, knows all other participants have voted commit and that, in time, participant itself will definitely commit.

State Transition Diagram for 3PC

(a) coordinator; (b) participant



3PC Protocol for Participant Voting Commit



```
                 Coordinator                                              Participant

Commit:
        write begin_commit to log
        send PREPARE to all participants                 ──────→   Prepare:
        wait for responses                                             write ready_commit to log
                                                         ←──────       send READY_COMMIT to coordinator
Pre_commit:                                                            wait for PRE_COMMIT or GLOBAL_ABORT
    if all participants have voted READY:
        write pre_commit to log
        send PRE_COMMIT to all participants              ──────→   Pre_commit:
        wait for acknowledgements                                      write pre_commit record to log
                                                         ←──────       send acknowledgement
Ready_commit:
    once at least K participants have acknowledged PRE_COMMIT:
        write commit to log                              ──────→   Global_commit:
        send GLOBAL_COMMIT to all participants                         write commit record to log
        wait for acknowledgements                                      commit transaction
                                                         ←──────       send acknowledgement
Ack:
    if all participants have acknowledged:
            write end_of_transaction to log
```

3PC Termination Protocols (Coordinator)

- Timeout in WAITING state
  - Same as 2PC. Globally abort transaction.
- Timeout in PRE-COMMITTED state
  - Write commit record to log and send GLOBAL-COMMIT message.
- Timeout in DECIDED state
  - Same as 2PC. Send global decision again to sites that have not acknowledged.

3PC Termination Protocols (Participant)

- Timeout in INITIAL state
  - Same as 2PC. Unilaterally abort transaction.
- Timeout in the PREPARED state
  - Follow election protocol to elect new coordinator.
- Timeout in the PRE-COMMITTED state
  - Follow election protocol to elect new coordinator.

3PC Recovery Protocols (Coordinator Failure)

- Failure in INITIAL state
  - Recovery starts commit procedure.
- Failure in WAITING state
  - Contact other sites to determine fate of transaction.
- Failure in PRE-COMMITTED state
  - Contact other sites to determine fate of transaction.
- Failure in DECIDED state
  - If all acknowledgements in, complete transaction; otherwise initiate termination protocol above.

3PC Recovery Protocols (Participant Failure)

- Failure in INITIAL state
  - Unilaterally abort transaction.
- Failure in PREPARED state
  - Contact other sites to determine fate of transaction.
- Failure in PRE-COMMITTED state
  - Contact other sites to determine fate of transaction.
- Failure in ABORTED/COMMITTED states
  - On restart, no further action is necessary.

3PC Termination Protocol After New Coordinator

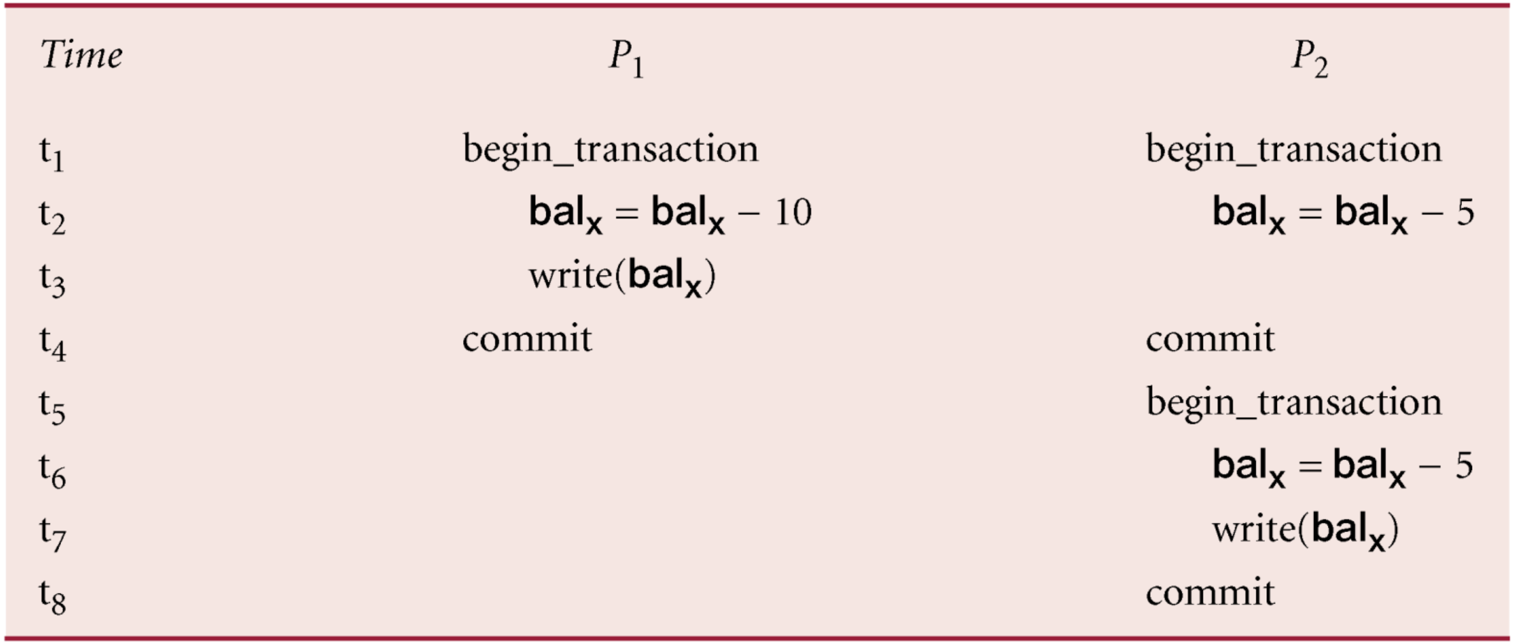
- Newly elected coordinator will send STATE-REQ message to all participants involved in election to determine how best to continue.

1. If some participant has aborted, then abort.
2. If some participant has committed, then commit.
3. If all participants are uncertain, then abort.
4. If some participant is in PRE-COMMIT, then commit. To prevent blocking, send PRE-COMMIT and after acknowledgements, send GLOBAL-COMMIT.

Network Partitioning

- If data is not replicated, can allow transaction to proceed if it does not require any data from site outside partition in which it is initiated.
- Otherwise, transaction must wait until sites it needs access to are available.
- If data is replicated, procedure is much more complicated.

Identifying Updates

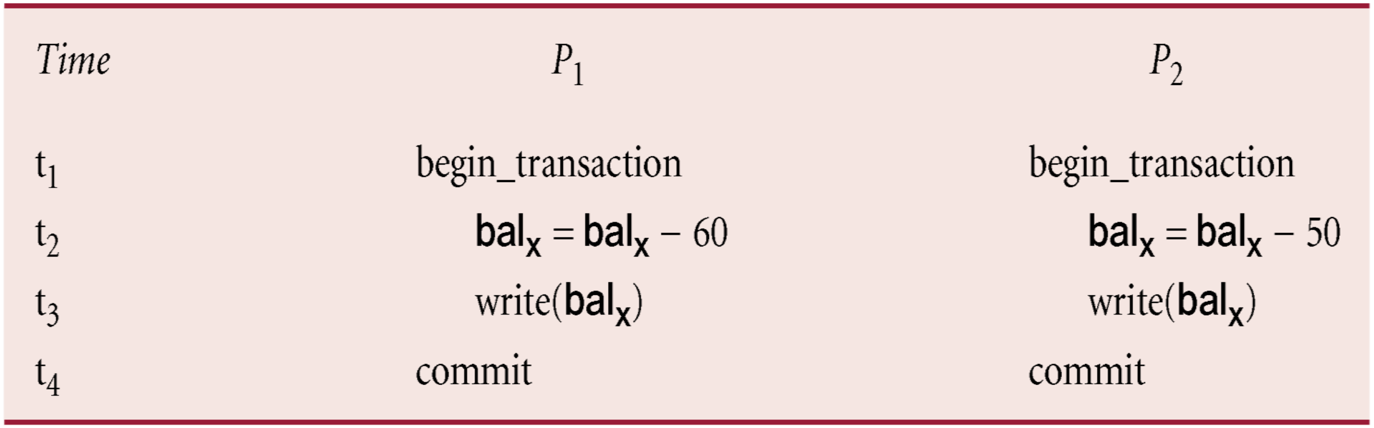


Time	P_1	P_2
t_1	begin_transaction	begin_transaction
t_2	$\mathbf{bal_x} = \mathbf{bal_x} - 10$	$\mathbf{bal_x} = \mathbf{bal_x} - 5$
t_3	write($\mathbf{bal_x}$)	
t_4	commit	commit
t_5		begin_transaction
t_6		$\mathbf{bal_x} = \mathbf{bal_x} - 5$
t_7		write($\mathbf{bal_x}$)
t_8		commit

Identifying Updates

- Successfully completed update operations by users in different partitions can be difficult to observe.
- In P_1, transaction withdrawn £10 from account and in P_2, two transactions have each withdrawn £5 from same account.
- At start, both partitions have £100 in bal_x, and on completion both have £90 in bal_x.
- On recovery, not sufficient to check value in bal_x and assume consistency if values same.

Maintaining Integrity



Time	P_1	P_2
t_1	begin_transaction	begin_transaction
t_2	$\text{bal}_x = \text{bal}_x - 60$	$\text{bal}_x = \text{bal}_x - 50$
t_3	write(bal_x)	write(bal_x)
t_4	commit	commit

Maintaining Integrity

- Successfully completed update operations by users in different partitions can violate constraints.
- Have constraint that account cannot go below £0.
- In P_1, withdrawn £60 from account and in P_2, withdrawn £50.
- At start, both partitions have £100 in bal_x, then on completion one has £40 in bal_x and other has £50.
- Importantly, neither has violated constraint.
- On recovery, bal_x is –£10, and constraint violated.

Network Partitioning

- Processing in partitioned network involves trade-off in availability and correctness.
- Correctness easiest to provide if no processing of replicated data allowed during partitioning.
- Availability maximized if no restrictions placed on processing of replicated data.
- In general, not possible to design non-blocking commit protocol for arbitrarily partitioned networks.