Transaction Processing
2-Phase-Commitment Protocol

Hans Philippi

March 23, 2020

Distributed transactions

Examples:

- A large bank managing accounts at different locations
- A money transfer between different banks
- A travelling agency, offering a combination of a hotel reservation and a flight reservation
- Choosing a date and time for a meeting of several people

Finishing distributed transactions

- Goal: reach unanimity concerning termination
- either *commit* or *abort*
- Atomic Commitment Protocol (ACP)
- complications by failures

Kinds of failures

- site failures

  site is operational or down (fail/stop behaviour)
- partial failure in network

  some sites operational, some sites down
- total failure

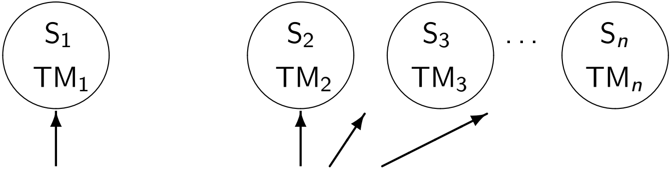
  all sites down
- communication failure

  network partition, non-communicaring components

assumptions

- undeliverable messages dropped
- failure detection by timeout

Two phase commit



S_1 TM_1

S_2 TM_2

S_3 TM_3 . . .

S_n TM_n

COORDinator PARTicipants

Generally, S_1 is homesite of this transaction

- every site maintains a DT-log (Distributed Transaction Log)
- every site will vote with respect to termination
  (`[ready T]` or `[abort T]`)
- decision by COORD: Commit or Abort

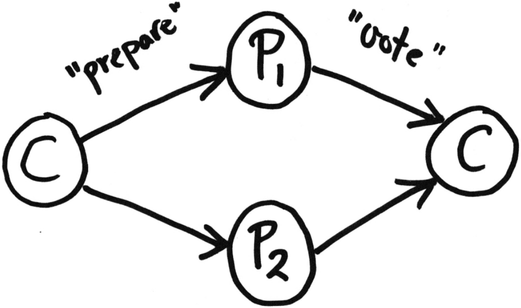
ACP: correctness requirements

- All processes that reach a decision, reach the same one
- A process cannot reverse its decision after it has reached one
- The Commit decision can only be reached if *all* processes vote `[ready T]`
- If there are no failures and all processes vote `[ready T]` then the decision will be Commit

Two phase commit: phase 1

1. The coordinator logs a <prepare-T> record
2. The coordinator sends a [prepare-T] message to all participants
3. When a participant receives a [prepare-T], it responds by sending a message containing that participant's vote to the coordinator
    - If the participant votes [abort T], it logs <abort T> and does a local rollback
    - Otherwise, it logs a record <Ready T> and sends a message [Ready T], in that order

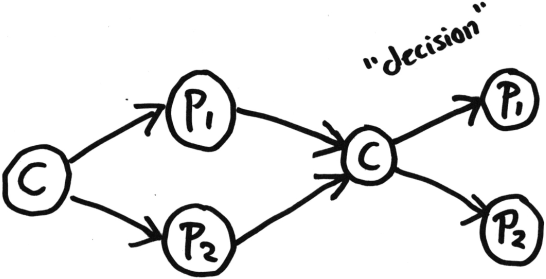
Two phase commit: phase 1



Two phase commit: phase 2

1. The coordinator collects the vote messages from all participants
    - If all of them are positive and the coordinator's vote is also positive, then the coordinator decides Commit, logs `<Commit T>` and sends `[Commit T]` messages to all participants, in that order
    - Otherwise the coordinator logs `<Abort T>` and sends `[Abort T]` messages to all participants that voted `[Ready T]`
2. Each participant that voted `[Ready T]` waits for a `[Commit T]` or `[Abort T]` message from the coordinator; when it receives it, it logs the decision and acts accordingly

Two phase commit: phase 2



Acting on time-out / recovery

So far, so good, but what to do when things go wrong?
Our point of view is the position of a participant (or coord), confronted with a time-out

- Apparently, some error occured, I have to start some recovery protocol
- I need to be aware of the transaction I am involved in
- I need to be aware of my status in the protocol
- The DT-log provides me with this status
- I act according to this status, with termination as primary goal, even if I have to force an ABORT

Acting on time-out / recovery

What do I see in my log?

- A `<START T>` record
- I am aware of the transaction I am involved in
- I know who are the participants

What do I see in my DT log? Five possibilities:

- A `<Commit T>` record
- An `<Abort T>` record
- A `<Prepare T>` record
- A `<Ready T>` record
- No `<Ready T>` record

Which are the corresponding actions?

Acting on time-out / recovery

What do I see in my DT log?

- A `<Commit T>` record, or
- An `<Abort T>` record

Action:

- Apparently, the decision has already been made
- Act according to this decision

Acting on time-out / recovery

What do I see in my DT log?

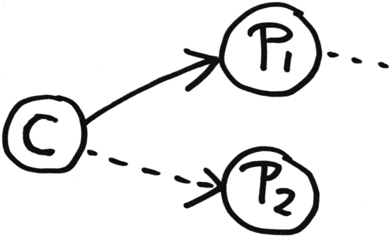
- A `<Prepare T>` record, and no decision record

Action:
?

Acting on time-out / recovery

What do I see in my DT log?

- A `<Prepare T>` record, and no decision record



Acting on time-out / recovery

What do I see in my DT log?

- A `<Prepare T>` record, no decision record

Action:

- My role was coordinator
- I started the protocol, but apparently no decision has been made yet
- I have the possibility to enforce an Abort
- I resume the protocol with an Abort decision

Acting on time-out / recovery

What do I see in my DT log?

- No `<Ready T>` record, and no decision record

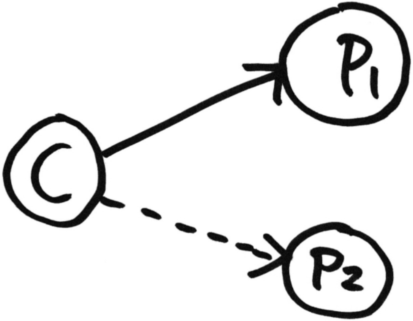
Action:

?

Acting on time-out / recovery

What do I see in my DT log?

- No `<Ready T>` record, and no decision record



Acting on time-out / recovery

What do I see in my DT log?

- No `<Ready T>` record, and no decision record

Action:

- My role was participant
- Apparently, I did not vote yet, so no Commit-decision has been made
- I have the possibility to enforce an Abort
- I resume the protocol with an Abort decision

Acting on time-out / recovery

What do I see in my DT log?

- A `<Ready T>` record, no decision record

Action:

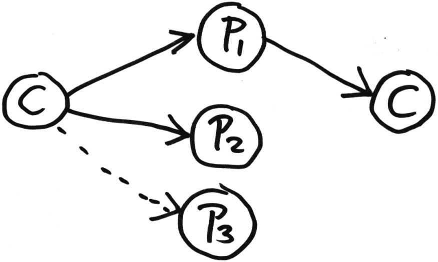
?

Acting on time-out / recovery

What do I see in my DT log?

- A `<Ready T>` record, no decision record

Action:



Acting on time-out / recovery

What do I see in my DT log?

- A `<Ready T>` record, no decision record

Action:

- My role was participant
- I voted positive, but I am not sure about the decision
- I try to contact the Coord and ask for the decision ...

Acting on time-out / recovery

What do I see in my DT log?

- A `<Ready T>` record, no decision record

Action:

- My role was participant
- I voted positive, but I am not sure about the decision
- I try to contact the Coord and ask for the decision ...
- If Coord does not respond, I start a Cooperative Termination Protocol (CTP)

Cooperative Termination Protocol

Run CTP (Cooperative Termination Protocol)

- Choose a Leader among the remaining participants
- This Leader acts as a new coordinator and requests the status from all remaining participants
- If one of the remaining participants knows the original decision, the Leader will broadcast this decision
- If one of the remaining participants has not yet voted, the Leader will broadcast an Abort decision
- If all of the remaining participants have voted positive and no one knows the decision, the protocol is *blocked!*

Blocking protocols

- 2PC is a blocking protocol
- blocking requires human intervention (from DBA) ...
- ... and/or correction protocols
- 3PC is an attempt to prevent blocking ...
- .. but cannot avoid blocking in the case of communication failures
- There is a proof of the claim that a non-blocking commitment protocol cannot exist