Distributed Data Storage and Management Part V

Saptarshi Pyne
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Jodhpur, Rajasthan, India 342030

CSL4030 Data Engineering Lecture 15
September 11th, 2023

What we discussed in the last class

- Commit protocols for distributed/global transactions ensure that a global transaction **either commits at all sites or aborts at all sites**.
  - The two-phase commit protocol (2PC)
  - The three-phase commit protocol (3PC)

The three-phase commit protocol (3PC)

SBI

Phase 1:

Same as that of 2PC.

Phase 2:

If and only if TC_{SBI} receives a “ready T_i” message from every TM before the timeout (*ready state*), TC_{SBI} sends a **“prepare_to_commit T_i”** message to all TMs. Otherwise, TC_{SBI} sends an “abort T_i” message to all TMs.

TC_{SBI} crashes in the process of sending the “prepare_to_commit T_i” or “abort T_i” messages to the TMs.

(If TC_{SBI} does not crash, Phase 3 will be similar to the remaining steps of 2PC.)

Phase 3:

If some of the TMs do not receive the “prepare_to_commit T_i” or “abort T_i” messages from TC_{SBI} before timeout, their TCs contact other available TCs. If at least a pre-specified number of TCs are up **(say, at least ‘k’ TCs are up)**, together they elect a new TC for this transaction (using an ‘election algorithm’).

TC_{new} checks whether at least one of the TMs have received a “prepare_to_commit T_i” message or not. If one of them did, TC_{new} sends a “commit T_i” message to all TMs. Otherwise, TC_{new} sends an “abort T_i” message to all TMs. Thus everything gets back on track.

Limitations:

High implementation complexity and network overhead. When there are network partitions or a large number of sites have failed, their TCs would not participate in the election of TC_{new}. Hence, TC_{new} needs to make sure that those sites honor TC_{new}’s decision of commit or abort once they are up again. For these reasons, 3PC is not widely used.

[1] Chap 19, Korth.

An alternative model to commit protocols: Persistent messaging

Can we mimic a cash transaction?

- A **persistent message** mimics cash
  - Persists site failures, message losses, and network partitions: **Guaranteed to be delivered exactly once**

Who handles the exceptions (e.g., the recipient account is closed)?

- The app developers
- Worth the effort if ‘blocking’ is abundant

Transactions vs. Workflows

A **workflow** is an action item that involves multiple transactions and, if required, human intervention.

E.g., processing a home loan application

- Entering the application record
- Gathering information about the credibility of the applicant from various external sources
- Accepting or rejecting the application

Today we discussed

- Persistent messaging: An alternative model to commit protocols

Remaining sub-topics for distributed databases

- Concurrency control with locking protocols
- Availability
  - High availability at the cost of consistency: The Cloud
- Multi-database systems for heterogeneous distributed databases
- Distributed directory systems for managing data
  - The lightweight directory access protocol (LDAP)

References

- A. SILBERSCHATZ, H.F. KORTH, S. SUDARSHAN (2011), Database System Concepts, McGraw Hill Publications, 6th Edition.
  - Chapter 19. Distributed Databases

- Paper: Bronson et al., “TAO: Facebook’s Distributed Data Store for the Social Graph”, 2013 USENIX Annual Technical Conference (USENIX ATC ‘13).
  - Video: https://www.usenix.org/conference/atc13/technical-sessions/presentation/bronson

Thank you