INM370 – Advanced Databases

Tutorial 7 – Distributed Transaction Management/ Concurrency Control/Recovery, and Replicated Databases

Model Answers

This week's tutorial is concerned with Transaction Management/Concurrency Control/Recovery in distributed databases, and it introduces replicated database systems.

In a distributed environment, locking based algorithms can be classified as centralized, primary copy or distributed. Compare and contrast these algorithms.

Question 1.

Answer. Locking is used as an implementation of concurrency control mechanism in distributed database system too (similarly to centralized database systems, which we have already covered).

- Centralized Locking
  - Single site that maintains all locking information.
  - One Lock manager for the whole of Distributed DBMS (DDBMS).
  - Local transaction managers involved in global transaction; they request and release locks from the Lock manager.
  - Or transaction coordinator can make all locking requests on behalf of local transaction managers.
  - Advantage – easy to implement.
  - Disadvantages – network bottlenecks (all transactions request locks from the central location), and worse reliability (single point of failure).
- Primary Copy
  - Lock managers distributed to a number of sites.
  - Each lock manager responsible for managing locks for a set of data items.
  - For replicated data item, one copy is chosen as primary copy, others are slave copies
  - Only need to "write-lock" primary copy of data item that is to be updated.
  - Once primary copy has been updated, change can be propagated to slaves.
    - But, only primary copy site is guaranteed to hold the current value(s).
  - Advantages – lower communication costs and better performance than Centralized 2PL.
  - Disadvantages – deadlock handling is more complex, since there's a degree of de-centralization in the system (wrt primary copies) - different from Centralised 2PC, where the central coordinator/single site has all lock info; slave copies updated eventually.
- Distributed
  - Lock managers distributed to every site.
  - Each lock manager responsible for locks for data at that site.
  - If data not replicated, equivalent to primary copy 2PL.
  - Otherwise, implements a Read-One-Write-All-(Available) – ROWA(A) – replica control protocol.
  - Using ROWA(A) protocol:
    - Any copy (if not failed) of replicated item can be used for read.
    - All copies must be "write-locked" before item can be updated.
  - Disadvantages - deadlock handling more complex; communication costs higher than primary copy 2PL.
- Majority Locking
  - Extension of Distributed 2PL.
  - To read or write a data item replicated at n sites, obtain a lock request from more than half the n sites where the item is stored.

- Transaction cannot proceed until majority of locks obtained.
- Overly strong in case of read locks, thus.
  - Any number of transactions can simultaneously hold a shared lock on a majority of the copies
  - But ONLY 1 transaction can simultaneously hold an exclusive lock on a majority of the copies

For further details see week 7 lecture slides, especially 6-14, and Section 25.2.3 from the Connolly and Begg book (6th Ed).

Describe the term "non-blocking" protocol, and explain how it is related to two- phase commit (2PC) protocol.

Question 2.

Answer: A non-blocking protocol should cater for both site and communication failures to ensure that the failure of one site will not affect processing at another site. In other words, operational sites should not be left blocked.

When 2PC is considered, in the event that a participant has voted COMMIT but has not received global decision and is unable to communicate with any other site that knows the decision, that site is blocked. Although 2PC has a cooperative termination protocol that reduces the likelihood of blocking, blocking is still possible and the blocked process will just have to keep on trying to unblock as failures are repaired.

Outline two alternative topologies to the Centralized topology of two-phase commit protocol (2PC).

Question 3.

Answer: The alternative topologies are *Linear* and *Distributed* 2PC.

There are different communication topologies (ways of exchanging messages) that can be employed to implement 2PC. The default one discussed in the lecture is the Centralised topology. In this topology all communication is funneled through the coordinator.

There is a number of improvements of protocol performance, such as linear and distributed topology. In linear topology, participants can communicate with each other as shown in the figure b) of the slide titled "2PC Topologies". The sites are ordered 1, 2…, n, where site 1 is the Coordinator and the remaining sites are participants. The 2PC protocol is implemented by a forward chain of communication from coordinator to participant n for the Voting phase, and a backward chain of communication for the Decision phase. In the voting phase, the coordinator passes the vote instruction to site 2, which votes and then passes its vote to site 3. Site 3 then combines its vote with that of site 2 and transmits the combined vote to site 4, and so on. When the n^{th} participant adds its vote, the global decision is obtained and passed backwards to participants n-1, n-2, and so on and eventually back to coordinator.

Although, linear 2PC incurs fewer messages than centralized 2PC, the liner sequencing does not allow for any parallelism.

"*Linear 2PC can be improved if the voting process adopts the forward linear chaining of messages while the decision process adopts the centralized topology, so that the site N can broadcast the global decision to all participants in parallel* ", p847-848 Connolly and Begg 6th Edition.

In Distributed 2PC, the coordinator sends the PREPARE message to all participants, which in turn send their decision to all other sites. Each participant waits for messages from the other sites before deciding whether to commit or abort the transaction. This in effect eliminates the need for the Decision phase of the 2PC, as the participants can reach a consistent decision, but independently.

For further details see the slide "2PC Topologies" in week 7 lecture, and the end of Section 25.4.3 from the Connolly and Begg book (6th Ed).

Explain the importance of data replication.

Question 4

Answer: Replication is an important mechanism, because it enables organizations to provide users with access to current data where and when they need it. It is intended to increase the fault tolerance of a system and contribute to improved reliability and availability, such that if one database/site fails another one can continue to serve clients' requests. It also can improve performance, e.g. by the replication scheme providing read access to data for any site (load balancing).

For further details, see week 7 lecture slides 29-32 and Section 26.1 from the Connolly and Begg book (6th Ed).