Chapter LXXVIII
SWIFT:
A Distributed Real Time Commit Protocol

Udai Shanker
M.M.M. Eng. College, India

Manoj Misra
IIT Roorkee, India

Anil K. Sarje
IIT Roorkee, India

INTRODUCTION

Many applications such as military tracking, medical monitoring, stock arbitrage system, network management, aircraft control, factory automation, and so forth that depend heavily on database technology for the proper storage and retrieval of data located at different remote sites have certain timing constraints associated with them. Such applications introduce the need for distributed real-time database systems (DRTDBS) [Ramamritham, 1993]. The implementation of DRTDBS is difficult due to the conflicting requirements of maintaining data consistency and meeting distributed transaction's deadlines. The difficulty comes from the unpredictability of the transactions' response times [Huang, 1991]. Due to the distributed nature of the transactions and in presence of other sources of unpredictability such as data access conflicts, uneven distribution of transactions over the sites, variable local CPU scheduling time, communication delay, failure of coordinator and cohort's sites, and so forth, it is not easy to meet the deadline of all transactions in DRTDBS [Kao & Garcia – Monila, 1995]. The unpredictability in the commitment phase makes it more serious because the blocking time of the waiting cohorts due to execute-commit conflict may become longer. Hence, due to unique characteristics of the committing transactions and unpredictability in the commitment process, design of an efficient commit protocol is an important issue that affects the performance of DRTDBS [Shanker, Misra & Sarje, 2006d].

BACKGROUND

The Two Phase Commit (2PC) is still one of the most commonly used protocols in the study of DRTDBS. Most of the existing commit protocols proposed in the literature, such as presumed commit (PC) and presumed abort (PA) [Haritsa, Ramamritham & Gupta, 2000] are based on it. Soparkar et al. [Nandit, Levy. Korth & Silberschatz, 1994] proposed a protocol that allows individual sites to unilaterally commit. If it is later found that the decision is not consistent globally then compensation transactions are executed to rectify errors. The problem with this approach is that many actions are irreversible in nature. The 2PC based optimistic commit protocol (OPT) [Gupta, Haritsa & Ramamritham, 1997] for real-time databases try to improve system concurrency by allowing executing transactions to borrow data from the transactions in their commit stage. This creates dependencies among transactions. If a transaction depends on other transactions, it is not allowed to start commit processing and is blocked until the transactions, on which it depends, have committed. The blocked committing transaction may include a chain of dependencies as other executing transactions may have data conflicts with it. Enhancement has been made in the Permits Reading of Modified Prepared-Data for Timeliness (PROMPT) commit protocol, which allows executing transactions to borrow data in a controlled manner only from the healthy transactions in their commit phase [Haritsa, Ramamritham & Gupta, 2000]. However, it does not consider the type of dependencies between two transactions. The abort of a lending transaction aborts all the transactions dependent on it. The impact of buffer space and admission control is also not studied. In case of sequential transaction execution model, the borrower is blocked for sending the workdone message and the next cohort can not be activated at other site for its execution. It will be held up till the lender completes. If its sibling is activated at another site anyway, the cohort at this new site will not get the result of previous site because previous cohort has been blocked for sending of workdone message due to being borrower [Shanker, Misra, Sarje & Shisondia, 2006c]. In shadow PROMPT, a cohort forks of a replica of the transaction without considering the type of dependency, called a shadow, whenever it borrows a data page.

The deadline-driven conflict resolution (DDCR) commit protocol maintains three copies of each modified data item (before, after and further) for resolving execute-commit conflicts [Lam, Pang, Son & Cao, 1999]. This not only creates additional workload on the system but also has priority inversion problems. Based on the concepts of above protocols [Lam, Pang, Son & Cao, 1999; Haritsa, Ramamritham & Gupta, 2000], Biao Qin and Y. Liu proposed a protocol Double Space (2SC) [Qin & Liu, 2003] which classifies the dependencies between lender and borrower into two types; commit and abort. The abort of a lending transaction only forces transactions in its abort dependency set to abort. The transactions in the commit dependency set of the aborted lending transaction continue as normal. However, 2SC creates inconsistency in case of write-write conflicts [Shanker, 2006e]. The protocols [Lam, Pang, Son & Cao, 1999; Qin & Liu, 2003] use blind write model whereas PROMPT uses update model.

SWIFT

A static two phase locking [Lam, Hung & Son, 1997; Lam, 1994] with higher priority (S2PL-HP) based distributed real time commit protocol named as SWIFT has been proposed.

Basic Idea of Protocol

A commit protocol can improve transaction success percentage by reducing the commit duration for each transaction, causing locks to be released