Localized-Access Protocols for Replicated Databases*

D. Agrawal  A. El Abbadi
Department of Computer Science
University of California
Santa Barbara, CA 93106

Abstract

In this paper, we present two protocols for efficient execution of transactions in replicated databases. Transactions are executed at a single site thus avoiding communication overhead and distributed commitment, which are required by most other replica control protocols. In the first protocol, data accessibility at a site can be dynamically reconfigured using special transactions, which are executed on demand. In the second protocol, data accessibility is reconfigured by migrating ownership of individual objects in the database. The two protocols present trade-offs with respect to atomicity, resiliency, and data availability. The approach of local execution of user transactions improves response time, eliminates the need for distributed commit protocols, and accommodates database heterogeneity.

1 Introduction

Data replication is traditionally used for increasing data availability and fault-tolerance in distributed systems. When a data object is replicated on several sites, the object may still be available for reading or writing even after failures have occurred. Thus data replication may increase data availability and hence increase the fault-tolerance of the database. On the other hand, replication also holds the promise of reducing communication costs incurred when a transaction can access locally stored data. If data is stored at the site where a transaction is initiated, operations can be executed locally, thus reducing both communication costs and the response time. However, local execution is

*This research is supported by the NSF under grant numbers CCR-8809387 and IRI-8809284.
often not possible since several sites may need to be accessed for replica synchronization. In this paper, we assume that sites are fail-stop [SS82] and communication links may fail to deliver messages. Combinations of such failures may lead to network partitioning [DGMS85]. We propose two schemes that employ replicated data primarily to eliminate remote data access by transactions, resulting in better performance in distributed databases.

Most replicated data management protocols have focussed on increasing data availability and fault-tolerance, often at the expense of communication costs. Several protocols are based on the quorum approach proposed by Gifford [Gif79] where both read and write operations must access several copies of an object, thus requiring communication between several sites. Few attempts have, however, been made to use replication to improve performance in distributed databases. The virtual partitions, the dynamic quorum, and the views protocols [ESC85, Her87, ET89a] address the issue of communication costs by always allowing a read operation to be executed by accessing a single copy. Write operations, on the other hand, must access several copies. The viewstamped replication protocol proposed by Oki and Liskov [OL88] extends the basic primary site approach [Sto79], and allows both read and write operations to be executed at a single site. However, before a transaction can commit it must communicate with a set of sites to inform them of its results. This protocol therefore reduces the overall response time of transactions, but does not effect the communication requirements. Kogan and Garcia-Molina proposed Bakunin networks [KGM87] where read operations can be executed locally at any site while write operations must be executed locally at specific, predefined sites. This approach reduces the communication costs of executing transactions and results in low response time. It, however, requires pre-analysis of the transactions in the database to determine which sites can support write operations.

More recently, several protocols have been proposed to eliminate the problem of distributed commitment that arises in distributed databases. Tam and Hsu [TH90] have used the idea of tokens to localize the access of data objects in distributed databases. Tokens are used to migrate data among the sites while guaranteeing data coherency. Special token transactions reliably manage the transitions of tokens. A special unilateral protocol is used to commit these token transactions. Soparkar and Silberschatz [SS90]