This paper presents an optimistic approach to transaction management for replicated databases. We propose a new transaction management protocol that guarantees global serializability and freedom from distributed deadlocks without relying on any properties of the DBMSs running at the local sites. In comparison to prior protocols, this protocol reduces the communication required to coordinate transactions by a factor of $r$, where $r$ is the average number of operations per transaction. We also consider implementation issues in reducing message overhead and discuss failure recovery.


Keywords: concurrency control, replication, distributed databases
1. Introduction

The management of replicated data in a distributed database is an old and well-studied problem. However, the classic solutions to the replicated data problem are applicable mainly to distributed systems with a relatively low transaction-processing rate. With distributed data warehouses and data marts at the high end, and distributed data in often-disconnected mobile computers at the low end [18], the problem of consistent access to replicated data with reasonably high transaction throughput represents a difficult challenge [9, 14, 15, 26, 28]. It was shown [14] that many proposed solutions do not scale up to meet the demands of systems that either require a high throughput rate, or a high degree of replication, or both. Furthermore, it does not appear that straightforward modifications of classic solutions will eliminate these deficiencies. Even 10 years after the publication of [14], the management of replicated data remains a vexing problem. The fundamental problem, as identified by [14], is that the standard transactional approach to the propagation of updates to replicas is unstable—deadlocks increase as the cube of the number of network sites and as the fourth power of transaction size. This is particularly problematic with relatively long data-mining queries and with mobile transactions. The former access many data items; while the latter effectively live for a long period of time. Thus, deadlock is no longer a rare event with a negligible effect on performance; instead, it is a barrier to the ability of systems to scale.

Several authors considered the problem of ensuring global serializability and atomicity without the use of an atomic commit protocol. They assumed that only transactions that execute at the primary site for a data item may initiate an update to that data item, and that propagation of updates to replicas occurs only after the update transaction has committed at the primary site [9, 14, 28, 5]. Such an approach was termed lazy-master.

To guarantee global serializability, the lazy-master approach must be augmented with one of the following:

- Restrictions on placement of primary copies of data among different sites [9].
- Restrictions on the order of replica updates propagation after the primary copy has been updated.
- A global concurrency-control mechanism that minimizes coordination among sites.

In this paper, we choose the lazy-master approach. We require that transaction executions be serializable and atomic, and that the transactions read only committed data. The atomicity protocol, however, should not cause blocking. We extend here the notion of lazy replica propagation and present an