AFOSR-TR. 83-0512

FINAL REPORT

CONCURRENT UPDATES AND RETRIEVAL
IN DISTRIBUTED DATABASE SYSTEMS

by

M. R. Stonebraker
E. Wong

Final Technical Report
July 1, 1981 - December 31, 1982
GRANT AFOSR-78-3596

ELECTRONICS RESEARCH LABORATORY
College of Engineering
University of California, Berkeley
94720

Approved for public release; distribution unlimited.
At its inception, this project was designed to represent a comprehensive program of research in the field of distributed database management. The problems to be dealt with were to include the three major topics in distributed database: query processing, concurrency control and crash recovery. In addition, the problem of interconnecting heterogeneous databases was also proposed. To a substantial extent, major progress has been achieved in all these areas. In this report a summary of the principal findings is presented.
1. Introduction

At its inception, this project was designed to represent a comprehensive program of research in the field of distributed database management. The problems to be dealt with were to include the three major topics in distributed database: query processing, concurrency control and crash recovery. In addition, the problem of interconnecting heterogeneous databases was also proposed.

To a substantial extent, major progress has been achieved in all these areas. In this report a summary of the principal findings is presented.

2. Query Processing

In a distributed DBMS, the database is fragmented, and the fragments distributed, with or without replication. For such systems it is convenient to classify queries into three categories:

(a) Local Queries that can be processed at a single site.
(b) Locally Processable Queries that can be processed at the sites in parallel without any need for intercommunication,
(c) Distributed Queries that do not fall into either of the above classes.

Queries of the first two classes can be processed with no data movement and require no strategies different from query optimization for a centralized database. However, processing of truly distributed queries entails both data movement and strategies that transcend centralized query optimization.
The first general strategy for processing distributed queries was formulated in [WONG77], and a strategy emphasizing parallelism was given in [EPST78]. As a part of the effort for this project, a reformulation that generalizes the above approaches was undertaken [WONG81]. In this formulation, query processing is viewed as an alternating sequence of data movement and local processing. Each operation in the sequence affects in one way or another the data available for processing at each site (collectively referred to as the "materialization" of the database). Query processing, then, can be formulated as a process of "dynamic re-materialization." Viewing the problem this has led to considerable progress on its solution.

Distributed query processing was also studied by Epstein and Stonebraker. In [EPST80] query processing experiments that were performed in a distributed database environment were reported. In this environment several algorithms were compared on the basis of number of bytes moved.

3. Concurrency Control

Locking is a fundamental technique for ensuring data integrity under concurrent accesses. In [RIES79] Ries and Stonebraker studied the problem of choosing the appropriate granularity for locking. The trade-off here is between excessive overhead (small granules) and reduced concurrency (large granules). In [CARE82] granularity hierarchies for locking are considered and several types of concurrency control algorithms are extended to take advantage of such hierarchies.

Multiple-copy consistency is another topic of major interest in distributed concurrency control. In [STON79], possible solutions to this problem are explored. In [CARE83] an abstract model of control
control algorithm is presented.

The model facilitates implementation-independent descriptions of various algorithms, allowing them to be specified in terms of the information that they require, the conditions under which blocking or restarts are called for, and the manner in which requests are processed. The model also facilitates comparisons of the relative storage and CPU overheads of various algorithms based on their descriptions. Results are given for single-site versions of two-phase locking, basic timestamp ordering, and serial validation. Extensions which will allow comparisons of multiple version and distributed algorithms are discussed as well.

4. Crash Recovery

Consistency in a distributed database system is based upon the notion of a transaction, a distributed atomic action. In [SKEE81, SKEE82], Skeen and Stonebraker studied commit protocols for preserving transaction atomicity (and hence consistency) in the presence of failures. They succeeded in:

(1) Introducing a formal framework for reasoning about the crash recovery problem.

(2) Showing fundamental limitations on the fault-tolerance of commit protocols.

(3) Deriving sufficient, and in many cases necessary, properties for a protocol to provide maximum fault-tolerance to various classes of failures.

(4) From the above properties, deriving families of fault-tolerant protocols.
Two failure classes are studied in detail: site failures and network partitioning.

In designing a commit protocol, the primary and overriding objective is to guarantee atomicity; the secondary objective is to maximize availability of the database. Since availability is limited if pending transactions must block (suspend execution) on failures, our focus is on nonblocking protocols.

The formal model introduced is based on nondeterministic finite state automata with failures viewed as a distinguished type of state transition. The model is used both in determining bounds on fault tolerance and in specifying and verifying the protocols summarized below.

Concerning site failures, the Nonblocking Theorem, yielding necessary and sufficient conditions for a commit protocol to be nonblocking was proved. From this result, a family of protocols (the three-phase protocols) was derived. These protocols never require an operational site to block on failures by other sites, even if the transaction coordinator fails.

Concerning site recovery, the nonexistence of nonblocking site recovery was conclusively proved.

Concerning network partitioning, the nonexistence of nonblocking solutions is again proved, and a family of protocols tunable toward maximizing the expected number of nonblocking sites is derived. These protocols are extremely resilient -- resilient when the cause of the failure or even its presence is in doubt.
5. **Heterogeneous Databases**

If a distributed database system is to integrate existing databases, then a potential problem is heterogeneity. The existing databases may differ in data model and in query language. In [KATZ80, KATZ82, KATZ83] several of the problems associated with heterogeneity were studied.

In [KATZ82] the problem of converting a program expressed in the CODASYL-DML (a procedural language) into a program written in a nonprocedural relational language was studied. The conversion process (dubbed "decompilation") is feasible only under certain circumstances and these are elucidated.

One of the side benefits of the decompilation study was the formulation of a data model (the access path model) that has also found application in physical database design [KATZ83].

6. **Implementation**

A major effort was undertaken to supplement a distributed version of INGRES. This effort is now complete and a multiple-machine version of INGRES is now operational on a local area network connecting three VAX processors.
REFERENCES


R.A.'s funded by Grant AFOSR-78-3596

K.P. Birman
D.S. Brunso
M.J. Cary
A. Guttman
S.M. Head
G.W. Mattinger
M.A. Meyer
F.R. Mueller
M. Murphy
R. Probst
J.K. Ranstrom
D.R. Ries
M.D. Skeen
M.A. Whyte
K.C. Wong
D.A. Wood
J.I. Woodfill
Publication Citations


M. Stonebraker, "Retrospection on a Data Base System," submitted to the ACM Transactions on Database Systems.

Publication Citations

Agrawal, Carey, Dewitt, "Deadlock Detection is Cheap," submitted to the SIGMOD Record, University of California, E.R.L. Memorandum No. UCB/ERL M83/5, 14 January 1983.


M. Carey, "Granularity Hierarchies in Concurrency Control," submitted to the 2nd SIGACT-SIGMOD Symposium on Principles on Database Systems, March 1983, Atlanta, GA.


