MIT/LCS/TR-207

ROBUST CONCURRENCY CONTROL FOR A DISTRIBUTED INFORMATION SYSTEM

Warren A. Montgomery

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Office of Naval Research under Contract No. N00014-75-C-0661

This blank page was inserted to preserve pagination.

ROBUST CONCURRENCY CONTROL FOR A DISTRIBUTED INFORMATION SYSTEM

by

Warren A. Montgomery

December, 1978

© Massachusetts Institute of Technology 1978

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Office of Naval Research under Contract No. N00014-75-C-0661.

Massachusetts Institute of Technology
Laboratory for Computer Science

Cambridge Massachusetts 02139

This empty page was substituted for a blank page in the original document.

ACKNOWLEDGEMENTS

Many people aided greatly in the production of this thesis. A complete acknowledgement would no doubt run longer than the work itself. My most sincere thanks to those who have not been included in this brief list, but none the less made my stay at MIT stimulating and productive.

I commend Professor L. Svobodova for her great endurance in reading many poorly written drafts of my early ideas and for her prompt response to each of my attempts to clarify those ideas. She was most helpful in that she could usually figure out what I meant to say, even when I was not sure myself, and yet still was able to read each draft as if she had not seen the work before. Dr. D. Clark was most helpful in providing key suggestions to clear up many of the vague areas, and his faith and interest in my early ideas were very much appreciated.

Professors F. J. Corbato and M. Hammer, my readers, provided many helpful suggestions for improving the thesis at and after my thesis examination. They were able to point out areas for improvement that were not so apparent to one who was caught up in the work.

I would like to thank Jim Gray of IBM Research, San Jose, and Dave Reed for several productive discussions of ideas in and related to my thesis. Such discussions were particularly helpful in providing early guidance on the merits of several of my ideas. Dave Reed must also be thanked for his help in creating many text-processing tools that were used in preparing the finished thesis.

I would like to thank all of the members of the Computer Systems Research group for providing an environment in which ideas were freely discussed and explored. They were also most helpful in being my agents after I had left M.I.T. I am extremely grateful for Allen Luniewski's help in relaying copies of the thesis between the printer at M.I.T., and myself in Illinois. My fellow workers and management at Bell Laboratories must also be thanked for their support in encouraging me to finish my thesis at a time during which it was most difficult to concentrate on the work.

My special thanks go to my wife, Carla, who endured five and a half long years of my career as a graduate student. She was always most understanding, even at times when her own thesis work was not going well. Her efforts in improving my writing have had a great impact on the quality and clarity of the final copy. Most of all, I would like to thank her for her unending confidence in my ability to finish, and for providing a strong incentive for doing so.

I would also like to thank the National Science Foundation for their support of my first three years at M.I.T. through the graduate fellowship program.

This empty page was substituted for a blank page in the original document.

ROBUST CONCURRENCY CONTROL FOR A DISTRIBUTED INFORMATION SYSTEM

by

WARREN A. MONTGOMERY

Submitted to the Department of Electrical Engineering and Computer Science
on December 1, 1978 in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy.

Abstract

This dissertation presents a collection of protocols for coordinating transactions in a distributed information system. The system is modeled as a collection of processes that communicate only through message passing. Each process manages some portion of the data base, and several processes may cooperate in performing a single transaction.

The thesis presents a model for computation in a distributed information system in which the sites and communication links may fail. The effects of such failures on the computation are described in the model. The thesis discusses implementation techniques that could be used to limit the effects of failures in a real system to those described in the model.

A hierarchical protocol for coordinating transactions is presented. The accesses to be performed during a transaction are pre-analyzed to select the protocols needed to coordinate the processes that participate in the implementation of the transaction. This analysis can be used to guide the organization of the data base so as to minimize the amount of locking required in performing frequent or important transactions. An important aspect of this mechanism is that it allows transactions that cannot accurately be pre-analyzed to be performed and correctly synchronized without severely degrading the performance of the system in performing more predictable transactions.

A novel approach to the problem of making updates at several different sites atomically is also discussed. This approach is based on the notion of a *polyvalue*, which is used to represent two or more possible values for a single data item. A polyvalue is created for an item involved in an update that has been delayed due to a failure. By assigning a polyvalue to such an item, that item can be made accessible to subsequent transactions, rather than remaining locked until the update can be completed. A polyvalue describes the possible values that may be correct for an item, depending on the outcome of transactions that have been interrupted by failures. Frequently, the most important effects of a transaction (such as the payment of money) can be determined without knowing the exact values of the items in the data base. A polyvalue for an item that is accessed by such a transaction may be sufficient to determine such effects. By using polyvalues, we can guarantee that a data item will not be made inaccessible by any failure other than a failure of the site that holds the item.

A strong motivation for the development of these protocols is the desire that the individual sites of a distributed information system fail independently, and that a site or a group of sites be able to continue local processing operations when a failure has isolated them from the rest of the sites. Many of the previous coordination mechanism have only considered the continued operation of the sites that remain with the system to be important. Another motivating factor for the development of these protocols is the idea that in many applications, the processing to be performed exhibits a high degree of locality of reference, in that most operations involve only a small number of sites. By structuring the coordination mechanism to take advantage of this locality of reference, one can have protocols that are simple, efficient, and robust for the particular application.

keywords: distributed data bases, synchronization, message passing systems, reliability.

CONTENTS

Acknowledgements 2

Abstract 3

Table of Contents 5

Table of Figures 8

1. Introduction 9

1.1. Reasons For Distribution 9
1.2. The Concurrency Control Problem in a Distributed Information System 12
1.3. Basic Assumptions and Goals 14
1.4. Related Work 18
1.5. Thesis Plan 26

2. The Process Model of Distributed Computing 29

2.1. The Model 29
2.2. Atomic Transactions Revisited 45
2.3. Summary 53

3. Atomic Broadcasting 55

3.1. Definitions 55
3.2. An Illustration of Atomic Broadcasting 58
3.3. A Mechanism for Atomic Broadcasting 59
3.4. Other Ordering Restrictions on Broadcast Messages 65
3.5. Implementation 68
3.6. Evaluation 80
3.7. Summary 85

4. Atomic Transactions in the Process Model 87

4.1. Analysis of Transactions 87
4.2. A Simple Approach to Transaction Synchronization 94
4.3. Classes of Transactions 99
4.4. A Hierarchical Scheme for Transaction Synchronization 103
4.5. Implementation of Hierarchical Locking 106
4.6. A Rejected Alternative Solution 119
4.7. Conclusions and Summary 122

5. Polyvalues: A Mechanism for Performing Atomic Updates to Distributed Data 125

5.1. Motivation (The Trouble with Locking) 125
5.2. The Polyvalue Mechanism for Avoiding Delay Due to Locking 132
5.3. Recovery of Pending Transactions 138
5.4. Use of Polyvalues in the Hierarchical Locking Scheme 142
5.5. Restricting the Spread of Polyvalues 148
5.6. Summary 150

6. Application of the Techniques to the Design of a Distributed Information System 151

6.1. The Problem 151
6.2. Analysis of the Transactions 155
6.3. Comparison with Other Mechanisms 168
6.4. Summary 171

7. Conclusions and Areas for Further Research 173

7.1. Summary of Thesis Work 173
7.2. Areas for Further Research 175
7.3. Summary 178

References 179

A. Proofs of the Protocols 182

A.1. Formalization of Atomic Broadcasting 182
A.2. Proof of Atomic Broadcasting 184
A.3. Correct Relative Sequencing of Broadcasts 186

B. An Analysis of the Propogation of Polyvalues 190

B.1. A Model for the Creation and Deletion of Polyvalues 190
B.2. Simulation of the Use of Polyvaues 195

Biographical Note 197

FIGURES

Figure 2.1	The Execution History of a Process	32
Figure 3.1	The Abbreviated Execution History of a Process	56
Figure 3.2	Non-Atomic Broadcasting	57
Figure 3.3	Coordinating Atomic Broadcasts with Message Forwarders	61
Figure 3.4	Moving a Process	73
Figure 3.5	A Physical Communication Topology	81
Figure 3.6	A Logical Topology for the Network of Figure 3.5	82
Figure 4.1	A Simple Transaction Graph	88
Figure 4.2	An Activity Graph For an Implementation of T	91
Figure 4.3	A Joint Activity Graph	93
Figure 4.4	A Transaction Using Delayed Locking	114
Figure 4.5	Concurrency Restrictions Due to Hierarchical Structure	121
Figure 5.1	A Two-Phase Commit Protocol	131
Figure 5.2	Recovery of Pending Transactions	143
Table 6.1	Transactions for Inventory Control	154
Figure 6.1	Transaction Graphs for Inventory Transactions	157
Figure 6.2	A Joint Transaction Graph of The Inventory Transactions	158
Figure 6.3	An Activity Graph for a Simple Data Base Organization	159
Figure 6.4	An Activity Graph for a more efficient Organization of the Data	161
Figure 6.5	A More Complete Activity Graph	162
Figure 6.6	An Activity Graph for a Redundant Data Base Organization	167
Table B.1	Typical Predictions of the Number of Polyvalues in a Database	194
Table B.2	Results of the Simulation of Polyvalues	196

Chapter 1
Introduction

Recent developments in electronic technology have made practical the interconnection of a large number of computer systems to form what I will refer to as a distributed information system. Each of the computer systems (or sites, as they are more frequently called) in the resulting system maintains some information and tools for accessing that information. The sites that make up a distributed information system may not be under the control of a single administrative authority. A distributed information system allows any user of any of the individual sites controlled access to the entire body of information managed by the system, while it allows each of the individual computer systems to control the use of the tools and information that it holds.

1.1 Reasons For Distribution

There are several good reasons for choosing such an organization for an information system rather than placing all of the information in a single large, shared computing facility. I will discuss some of these reasons briefly.

1.1.1 Autonomy

A very important reason for choosing a distributed organization for an information system is the autonomy of the individual sites. A recent study [DOliveira77] has shown that the ability to partition the authority and responsibility for information management in a distributed system is the most important reason for many businesses considering distributed

information systems. In a distributed system, each site has control over the information that it manages, and can set its own policies for controlling the availability of that information. As we shall see, autonomy has important implications for the assumptions that can be made about the cooperation of individual sites in the execution of processing operations, and for the protocols that can be used to coordinate such operations.

1.1.2 Reliability

A second reason for distribution is reliability. There are two ways in which a distributed information system can be made more reliable than a central facility. One way to achieve greater reliability in a distributed system is to replicate information, storing it at two or more of the sites in a distributed system. Replication increases the availability of information in a system with unreliable sites. A single failure does not make replicated information inaccessible. Unfortunately, modifying replicated information is much more difficult than modifying non-redundantly stored information. While a great deal of research has gone into the development of protocols to update replicated data, the problem remains difficult, and such updates are costly in that they require extensive communication between sites, reducing the economic advantage of distribution.

A second source of increased reliability, and one which I consider to be much more important, is the the failure of a single site or communication link does not necessarily make the entire system fail, while in a single, centralized system, the failure of a single component frequently interrupts all processing in progress. The individual sites in a distributed information system will be smaller and simpler than a single large computer system with storage and processing power equivalent to the total of that of the individual sites. This simplicity should mean that the sites in a distributed system fail less frequently than the single machine of a centralized system. Thus if a distributed system can be constructed so as

to limit the effects of a failure at one site to the interruption of processing that requires information at that site, the reliability of a distributed information system as seen by any individual user will be substantially better than that of a single shared machine.

1.1.3 Economics

A third reason for distribution is an economic advantage that makes a group of small computer systems less costly to manufacture than an "equivalent" single large machine. A single computer with a certain processing rate and storage capacity costs substantially more than a collection of smaller machines with the same aggregate processing rate and storage size. In addition to the computing hardware, communication and software development contribute to the cost of a distributed information system. Frequently, the information to be managed can be partitioned in such a way that most of the processing operations do not require information from more than one of the partitions. Each partition can be assigned to a small computer system capable of performing the processing required for the information in that partition. The cost of communication between sites in such a system would be relatively small. If the extra cost of developing software for a distributed information system can be kept small, a distributed information system may be substantially less costly than an equivalent central facility.

1.1.4 Flexibility

A fourth reason for distribution is flexibility. Changes in the amount of information to be managed by the system can require increasing or decreasing the storage and processing capacity. In a central system, this may require replacing the entire machine with one of a different capacity. In the distributed system, capacity changes can frequently be

accomplished by adding or deleting sites, with minimal impact on the sites not being changed.

Consider, for example, a corporation that has just acquired a subsidiary, and needs to modify its administrative information management system to manage the new subsidiary. Merging the information management systems of the parent company and the subsidiary into a single central facility could be very difficult. If the information management system being used by the corporation is distributed, however, the merger can be accomplished by adding one or more sites to manage the subsidiary.[1]

1.2 The Concurrency Control Problem in a Distributed Information System

Several problems must be overcome in order to make a distributed information system as easy to use as a central facility. The subject of this thesis, and what I believe to be the most difficult of these problems, is controlling the sequencing of user specified processing operations. The result of performing such processing operations concurrently should be the same as that obtained by performing them in some sequential order. Before this problem can be discussed in detail, we must have a more precise definition of the way in which stored information can be manipulated. For this purpose, I adopt terminology that has commonly been used in data base systems.

---

1. In rare cases, the existing information systems of the parent and the subsidiary may be