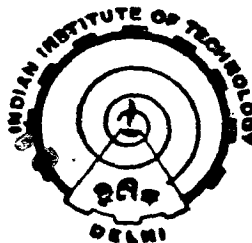


**SOME STUDIES ON
CONCURRENCY CONTROL AND RECOVERY
IN DISTRIBUTED DATABASE SYSTEMS**

By
RADHA RAMAN SINHA

Thesis submitted in fulfilment
of the requirements of the degree of
DOCTOR OF PHILOSOPHY



Department of Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY, DELHI

1988

I. T. DELHI
LIBRARY
Acc. No. TH-1608

dt. 21.8.88

[Handwritten signature]

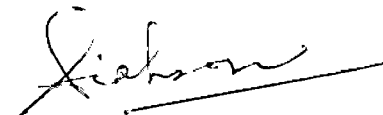
TH
681.3-5
SIM - 5



IN FOND MEMORY OF
MY FATHER WHO ALWAYS
ADVOCATED HONESTY AND
DETERMINATION FOR WORK

C E R T I F I C A T E

This is to certify that the thesis entitled "SOME STUDIES ON CONCURRENCY CONTROL AND RECOVERY IN DISTRIBUTED DATABASE SYSTEMS" being submitted by Radha Raman Sinha for the award of the degree of DOCTOR OF PHILOSOPHY to the Indian Institute of Technology, Delhi is a record of bonafide research work he has carried out under my supervision. The results contained in this thesis have not been submitted to any other University or Institute for award of a degree or diploma.


(S. I. Ahson)

Professor

Department of Electrical Engineering
Indian Institute of Technology, Delhi
NEW DELHI - 110016

A C K N O W L E D G E M E N T S

I express my deep sense of gratitude and appreciation to my supervisor Professor S.I.Ahson for his invaluable guidance, useful discussions and critical suggestions during the course of this research work.

I am thankful to Dr. Mukul K Sinha, Director, Expert Software Consultants Private Limited for his invaluable suggestions and help in shaping up this thesis.

Lastly thanks are due to Shri Arvind Kanaujia for typing the manuscript.

I.I.T. DELHI

JULY, 1988



RADHA RAMAN SINHA

ABSTRACT

A two-step commit algorithm which assures transaction update atomicity in a distributed database is modelled using Bipolar Synchronization schemes. The modelling helps in determining the numerous crash possibilities and in specifying recovery procedures and associated protocols. The use of unalterable logs is required and modelled accordingly.

A new scheme for maintaining multicopy data has been presented. Replicas of a data item, stored at various sites, are structured into one or more disjoint groups named representation trees. The scheme is more general and by restructuring the representation trees it is shown that : (1) the primary copy approach (2) the majority copies approach (3) the true copy token approach and (4) the quorum approach result as its special cases.

Some number of votes are assigned to each node_replica of the representation trees. Root_Replica represents the whole representation tree, and effectively has the cumulative sum of votes assigned to each node_replica (including the root) contained in the tree. A child has no right to vote till the Parent replica is active. A transaction scans only the root_replicas for collecting votes, resulting in less number of messages than that required in the basic quorum scheme.

Since the scheme presents a generalized and integrated view of multi_copy data, it provides a tool for comparing various schemes. By organizing the structure of the representation trees, by assigning different votes to replicas, and by tuning a read and write quorum the database administrator can control the reliability and performance characteristics most suited to the application. This scheme offers a performance improvement without degrading the resiliency in case of failures. It also provides increased concurrency as updates of non root_replicas are done in parallel, and are not involved in actual transactions.

CONTENTS	PAGE NO
ACKNOWLEDGEMENTS i
ABSTRACT ii
LIST OF FIGURESviii
CHAPTER-1 INTRODUCTION 1
1.1 Modelling distributed databases 3
1.2 Hierarchical Structure for Replicated Data bases 3
1.3 Thesis organization 4
CHAPTER-2 MODELLING TWO-PHASE COMMIT PROTOCOL USING BP SCHEME 6
2.1 Introduction 6
2.2 Two-Phase Commit Protocol 7
2.3 A Petri-Net Model for Two Normal Sites 10
2.4 A Petri-Net extended to a Simple Crash 12
2.5 BP Schemes 13
2.5.1 Firing Rules 15
2.5.2 Dead lock 17
2.5.3 An Interpreted B P Scheme 17
2.6 A BP Scheme Model for Two Normal sites 20
2.7 General Model for N Slaves 24
2.8 General Model for N Messages at Master Site 26
2.9 Model for a Simple Crash 28
2.10 Conclusions 31

CHAPTER-3 MODELLING CRASHES AND RECOVERY PROCEDURES	32
3.1 Introduction	32
3.2 The Effect of Crashes on Message Transmission	32
3.3 Recovery Procedures	34
3.4 Detailed Modelling for Recovery	39
3.4.1 Model for Autonomous Recovery	39
3.4.2 Model for Dependent Recovery	46
3.5 Well Behavedness of the B P Scheme Model	53
3.6 Comparative Study with a Model Based on Coloured Petri-Net	57
3.7 Conclusions	58
CHAPTER-4 CONCURRENCY CONTROL FOR REPLICATED DATA:A REVIEW	59
4.1 Introduction	59
4.2 Correctness for Replicated Data	65
4.2.1 The Write-All Approach	66
4.2.2 The Write-All Available Approach	67
4.3 Related Works	69
4.3.1 Available Copies Algorithm	69
4.3.2 The Primary Copy Replicated Data Model	75
4.3.3 The True Copy Token Replicated Data Model	76
4.3.4 The Majority Consensus Replicated Data Model	77
4.3.5 The Quorum Based Replicated Data Model	77

	4.3.6 The Directory Oriented Available Copies Replicated Data Model	79
	4.4 Conclusions	80
CHAPTER-5	HIERARCHICAL STRUCTURE VOTING FOR MULTICOPY DATA	82
	5.1 Introduction	82
	5.2 The Distributed System Architecture	83
	5.3 The Basic Replicated Data Model	83
	5.4 The Transaction Processing Model	86
	5.5 The Hierarchical Replicated Data Model	87
	5.5.1 The Hierarchical Replicated Structure	87
	5.5.2 The Hierarchical Replicated Data Model	90
	5.5.3 The Processing of a Transaction	91
	5.6 The Hierarchical Replicated Data Model: A Unifying View	93
	5.6.1 Modelling Primary Copy Replicated Data Model	93
	5.6.2 Modelling Majority Consensus Replicated Data Model	93
	5.6.3 modelling quorum based replicated data Model	93
	5.6.4 modelling true copy token Replicated Data Model	94
	5.7 Conclusions	94
CHAPTER-6	HANDLING OF FAILURES AND RECOVERY	96
	6.1 Introduction	96

6.2	Link Failures	98
	6.2.1 Leaf Link Failure	98
	6.2.2 Root Link Failure	98
	6.2.3 Intermediate Link Failure	103
	6.2.4 Site Failures	105
	6.2.4.1 Leaf Replica Site Failure	105
	6.2.4.2 Node Replica Site Failure	105
6.3	Recovery	105
	6.3.1 Recovery of Link	105
	6.3.1.1 Link Failure Ignored	106
	6.3.1.2 Attached to Other root_replica	106
	6.3.1.3 Declared independence unilaterally	107
	6.3.1.4 Attach to Other node_replica at the same Level.	107
	6.3.1.5 Connected to grandparent_replica	107
	6.3.1.6 Recovery of Site	108
6.4	Transaction Procedures	108
	6.4.1 Introduction	108
	6.4.2 The delink_attach Transaction	109
	6.4.3 The create_root Transaction	111
	6.4.4 The link grandparent Transaction	113
	6.4.5 The merge_root Transaction	113
	6.4.6 The link_oldparent Transaction	114
6.5	Conclusions	115

CHAPTER-7	SUMMARY AND SUGGESTIONS FOR	FUTURE WORK	116
	7.1	Summary	116
	7.2	Suggestions for Future Work	118
REFERENCES			120
APPENDIX			125