

AIN SHAMS UNIVERSITY
Faculty Of Engineering
Computers And Systems Engineering Department

IMPLEMENTATION OF DISTRIBUTED DATABASE ENHANCEMENT FUNCTIONS

BY

HESHAM EL-SAID MOHAMED
(B.Sc. Computer Eng. - Military Technical college)

A Thesis Submitted in Partial Fulfillment of
the Requirement for the Degree of
MASTER OF SCIENCE
IN COMPUTER ENGINEERING

Supervised by

Prof. Dr. M. A. R. Ghonaimy
Professor of computer systems
Faculty of engineering - Ain Shams University

Dr. Medhat Fakhry
Dr. of computer systems
Military Technical college

CAIRO, EGYPT
1997

بسم الله الرحمن الرحيم

قالوا سبحنك لا علم لنا الا ما علمتنا انك انتم العليم العظيم

صلى الله العظيم



Approval sheet

Thesis title:

**IMPLEMENTATION OF DISTRIBUTED DATABASE
ENHANCEMENT FUNCTIONS.**

Presented by:

Hesham El-Said Mohamed.

Examiner Committee.

Name

Signature

1- Prof. Dr. Ibrahim Farag Easa.

**Dean, Institute of Statistical Studies and Research.
Cairo University.**



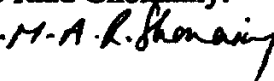
2- Prof. Dr. Hani Mohamed Kamal Mahdi.

**Prof. of Computer System.
Faculty of Engineering.
Ain shams University.**



3- Prof. Dr. Mohamed Adeeb Riad Ghonaimy.

**Prof. of Computer System.
Faculty of Engineering.
Ain Shams University.**



Date / / 1997

SUMMARY

The aim of this research is to study the effect of distributed processing systems (client/server architecture) on the deployment of database management software on different processors. In this thesis we start by studying the distributed database approach, which is a collection of data that belongs logically to the same system but is physically spread over the sites of a network. The architecture of a distributed database system consists of multiple computers, network, data processor software for local data management, application processor software for distributed functions and communication software. Distributed database systems (DDBSs) have a lot of advantages, they improve performance, reliability, availability, shareability, and expandability. DDBSs also give a local autonomy and they are more economical from the communication cost point of view. DDBSs have also some disadvantages, complexity, lack of experience, cost of communication and additional software to manage the distribution and difficulty of change between already existing centralized databases to distributed databases. There are two approaches for developing distributed databases, bottom-up integration and top-down distribution. There are a lot of data distribution strategies, basic distribution relation, data fragmentation (horizontal, vertical and mixed fragmentation), and replicated relations (full replication, no replication and partial replication). We show the transition of DDBSs to the client/server approach and how client/server features favor its use. We show the problems associated with distributed database systems, from administration point of view (catalog management, object naming, authorization, protection and archival check points), from transaction management point of view (two phase commit and three phase commit), from concurrency control point of view (locking and dead lock), from query optimization point of view (join sequencing, fragment access optimization and replica access optimization), and from integrity maintaining point of view (fragment integrity constraints and distributed integrity constraints). We analyze the different modes of distributed processing, host/terminal mode, network/file-server mode and client/server mode. We consider hardware components, systems services and application architecture of each mode in this analysis. We show the components, characteristics and communication aspects of the client/server environment. We

analyze the different types of database server configurations for user connections (single-task server, dedicated server and multithreaded server). We show that single-task server mode is used for nonnetworking (host based database server systems), dedicated server mode is used for administration tasks, and multithreaded server mode is widely used for client/server database environment. We show the distribution of the client/server database system processes between the clients and the server, and how they are interacting among the network. We develop a case study to establish client/server database environment including configuring conversational middleware for a database in client/server environment, starting middleware network listener process to accept and route remote client connections to the database server and configuring the database server as a multithreaded server. We show how to establish client/server database application, including (the role of the administrator on behalf of the application, schema design aspects and creating application schema aspects). We develop a case study for establishing and creating client/server database schema. Finally we study the query execution strategies (data shipping and query shipping) in client/server database environment, including the execution plan for each strategy. We show recommended guide lines to improve query execution performance for client/server database environment. The study concluded by studying query processing strategies and applying tuning guide lines.

ABSTRACT

HESHAM EL SAID MOHAMED
IMPLEMENTATION OF DISTRIBUTED DATABASE
ENHANCEMENT FUNCTIONS

Faculty of engineering - Ain Shams University
Dep. of computers and systems engineering

Client/server environment is now the most used environment of the distributed processing. Most of the distributed database management systems (DDBMS) is directed to this environment. There are a new view of establishing and tuning the distributed database (DDB) according to this environment. A middleware can be used to make the connection between client's applications and the database servers transparent. Overall system performance can be enhanced through controlling database management system (DBMS) use of space. There are two query execution strategies, query shipping (used with relational database) and data shipping (used with object oriented database). The query performance can be enhanced by tuning the server cache memory, reducing disk contention, tuning the query structure query language (SQL) statements and centralizing application logic at the server.

KEYWORDS

Buffer cache, Client/server, Data shipping, Dictionary cache, Distributed database, Library cache, Message passing, Query execution, Query execution performance, Query shipping, Remote procedure call, Stored procedure, Triggers, Storage parameters, Tuning cache memory.

STATEMENT

This thesis is submitted to Ain Shams University for the degree of Master of Science in Civil Engineering.

The work included in this thesis was carried out by the author in the Department of Computer and Systems Engineering, AIN Shams University, from May, 1995 to June, 1997.

No part of this thesis has been submitted for a degree or a qualification at any other University or institution.

Date June 30, 1997
Signature
Name Hesham El-said Mohamed

VITA

NAME	HESHAM EL-SAID MOHAMED
DATE OF BIRTH	26/9/65
SCIENTIFIC DEGREE	B Sc IN COMPUTER ENGINEERING MILITARY TECHNICAL COLLEGE
DATE OF GRADUATION:	JULY, 1986
CURRENT JOB	OFFICER IN ARMY FORCES

ACKNOWLEDGMENT

I would like to express my thanks and obligation to whom helped me, that without their help the completion of this work was impossible.

I wish to express my deep gratitude to Prof. Dr. M. Adeeb R. Ghonaimy, who supervised this study and guided its progress. I wish to thank him for his kind supervision, useful suggestions, and constructive criticism.

I am deeply thank Dr. Medhat Fakhery, who supervised this study and guided it progress. I wish to thank him for his kind supervision, useful suggestions, and constructive criticism.

TABLE OF CONTENTS

CHAPTER ONE

DISTRIBUTED DATABASE APPROACH PRIOR TO THE CLIENT/SERVER ARCHITECTURE	1
1.1 DEFINITION OF DISTRIBUTED DATABASE	1
1.1.1 ARCHITECTURE OF A DISTRIBUTED DATABASE SYSTEM	1
1.1.2 WHY DISTRIBUTED DATABASE MANAGEMENT SYSTEMS	2
1.1.3 ADVANTAGES AND DISADVANTAGES OF DBMSs	3
1.2 DISTRIBUTED DATABASE VS. DISTRIBUTED PROCESSING	4
1.3 APPROACHES FOR DEVELOPING DISTRIBUTED DATABASE	6
1.3.1 BOTTOM-UP INTEGRATION	6
1.3.2 TOP-DOWN DISTRIBUTION	6
1.4 DISTRIBUTED DATABASE NEEDED CAPABILITIES	12
1.4.1 FROM THE LOCAL AUTONOMY POINT OF VIEW	12
1.4.2 FROM THE DISTRIBUTION TRANSPARENCY POINT OF VIEW	13
1.4.3 FROM THE HETEROGENEITY POINT OF VIEW	14
1.5 DISTRIBUTED DATABASE TRANSITION TO CLIENT / SERVER APPROACH	15
1.5.1 CLIENT / SERVER CHARACTERISTICS	15
1.5.2 IMPLEMENTING DBMS IN CLIENT/SERVER	17

CHAPTER TWO

PROBLEMS ASSOCIATED WITH DISTRIBUTED DATABASE	19
2.1 DISTRIBUTED DATABASE ADMINISTRATION	19
2.1.1 CATALOG MANAGEMENT IN DISTRIBUTED DATABASE	19
2.1.2 DISTRIBUTED OBJECT NAMING	22
2.1.3 AUTHORIZATION AND PROTECTION	22
2.1.4 ARCHIVAL CHECKPOINTS	23
2.2 DISTRIBUTED TRANSACTION MANAGEMENT	24
2.2.1 PROPERTIES OF TRANSACTIONS	24
2.2.2 TWO PHASE COMMIT	26
2.2.3 DISTRIBUTED TWO PHASE COMMIT	27
2.2.4 THREE PHASE COMMIT	29

2.3	DISTRIBUTED CONCURRENCY CONTROL	30
2.3.1	CONCURRENCY CONTROL BASED ON LOCKING IN CENTRALIZED DATABASES	30
2.3.2	CONCURRENCY CONTROL BASED ON LOCKING IN DISTRIBUTED DATABASES	32
2.3.3	DISTRIBUTED DEADLOCKS	33
2.3.4	DEADLOCK DETECTION USING CENTRALIZED OR HIERARCHICAL CONTROLLERS	35
2.3.5	DECENTRALIZED DEADLOCK DETECTION	37
2.3.6	DISTRIBUTED DEADLOCK AVOIDANCE	38
2.4	DISTRIBUTED QUERY OPTIMIZATION	39
2.4.1	JOIN SEQUENCING	40
2.4.2	FRAGMENT ACCESS OPTIMIZATION	41
2.4.3	REPLICA ACCESS OPTIMIZATION	41
2.4.4	OPTIMIZATION IN A MULTIDATABASE SYSTEM ENVIRONMENT	42
2.4.5	CENTRALIZED VERSUS DISTRIBUTED QUERY COMPILATION	42
2.5	DISTRIBUTED INTEGRITY MAINTAINING	43
2.5.1	FRAGMENT INTEGRITY	43
2.5.2	DISTRIBUTED INTEGRITY CONSTRAINTS	43
 CHAPTER THREE		
CLIENT/SERVER (DATABASE MANAGEMENT/ COMMUNICATION) ENVIRONMENT		45
3.1	TRANSITION FROM TRADITIONAL DATABASE MANAGEMENT TO CLIENT/SERVER DATABASE MANAGEMENT	45
3.1.1	HOST / TERMINAL PROCESSING MODE	45
3.1.2	STAND ALONE / PCs PROCESSING MODE	48
3.1.3	NETWORK / FILE SERVER PROCESSING MODE	49
3.1.4	CLIENT/SERVER PROCESSING MODE	52
3.2	THE COMPONENTS OF CLIENT/SERVER ENVIRONMENT	55
3.3	COMMUNICATION ASPECTS IN CLIENT/SERVER MODE	57
3.3.1	MESSAGE PASSING-BASED TECHNIQUES	58
3.3.2	REMOTE PROCEDURE CALLS	61
3.4	NETWORKED SQL	63
3.4.1	SQL API	64
3.4.2	SQL INVOKED VIA AN API	65
3.4.3	INTEROPERABILITY	65
3.4.4	PERFORMANCE ASPECTS	66

3.5 SOME MODERN CLIENT/SERVER DBMS ARCHITECTURES	68
3.5.1 CLIENT/SERVER ARCHITECTURE	68
3.5.2 RAD-UNIFY TYPE OF DBMS ARCHITECTURE	69
3.5.3 ENHANCED CLIENT/SERVER ARCHITECTURE	70
3.6 TYPES OF DATABASE SERVER CONFIGURATIONS	72
3.6.1 SINGLE-TASK SERVER MODE	74
3.6.2 DEDICATED SERVER MODE	73
3.6.3 MULTITHREADED SERVER MODE	74
 CHAPTER FOUR	
PROBLEM FORMULATION: FRAMEWORK FOR A CLIENT/SERVER DATABASE CASE STUDY	77
 4.1 PROBLEM DEFINITION	77
4.2 THE CLIENT/ SERVER DATABASE CASE STUDY	79
4.3 THE CLIENT / SERVER DATABASE ENVIRONMENT	80
4.3.1 CLIENT PROCESSES	81
4.3.2 SERVER PROCESSES	82
4.3.3 MIDDLEWARE FOR CLIENT/SERVER DATABASE	83
4.4 CLIENT/SERVER DATABASE ADMINISTRATION	84
4.5 CLIENT/SERVER DATABASE QUERY PROCESSING OPTIMIZATION	85
 CHAPTER FIVE	
ESTABLISHING AND TUNING CLIENT/SERVER DATABASE (ENVIRONMENT-APPLICATION)	87
 5.1 ESTABLISHING AND TUNING CLIENT/SERVER DATABASE ENVIRONMENT	87
5.1.1 SETTING THE SERVER OPERATING SYSTEM FOR THE CLIENT/SERVER ENVIRONMENT	88
5.1.2 CREATING AND EDITING DATABASE INITIALIZATION PARAMETER FILE	88
5.1.3 CREATING A NEW DATABASE AND STARTING THE DATABASE SERVER	89
5.1.4 CONFIGURING THE DATABASE SERVER FOR THE CLIENTS CONNECTION	89
5.1.5 CONFIGURING CONVERSATIONAL MIDDLEWARE FOR DATABASE IN CLIENT / SERVER ENVIRONMENT	90
5.1.6 STARTING THE NETWORK LISTENER	94
5.1.7 CONFIGURING THE DISPATCHER AND SERVER PROCESSES	94

5.2 ESTABLISHING AND TUNING CLIENT/SERVER DATABASE APPLICATION	95
5.2.1 DATABASE ADMINISTRATION ON BEHALF OF APPLICATIONS	96
5.2.2 DESIGNING AN APPLICATION SCHEMA	96
5.2.3 CREATING THE APPLICATION SCHEMA	102
5.2.4 CASE STUDY FOR ESTABLISHING CLIENT/SERVER DATABASE SCHEMA	115
 CHAPTER SIX	
STUDY OF QUERY EXECUTION STRATEGIES FOR CLIENT/SERVER DATABASE SYSTEMS	116
6.1 INTRODUCTION	116
6.2 EXECUTION PLANS	117
6.3 QUERY EXECUTION POLICIES	118
6.3.1 DATA SHIPPING POLICY	119
6.3.2 QUERY SHIPPING POLICY	120
6.4 IMPROVING QUERY EXECUTION PERFORMANCE FOR CLIENT / SERVER DATABASE ENVIRONMENT	122
6.4.1 USING MEMORY CACHES	122
6.4.2 REDUCING DISK CONTENTION	128
6.4.3 REDUCING NETWORK I/O BY MOVING THE APPLICATION LOGIC TO THE DATABASE SERVER	130
6.5 CASE STUDY FOR QUERY PROCESSING STRATEGIES	136
6.5.1 QUERY SHIPPING POLICY	136
6.5.2 EXPLAINING STATEMENT EXECUTION TO DIAGNOSE TUNING PROBLEMS	138
6.5.3 DATA SHIPPING POLICY	138
6.5.4 TUNING MEMORY CACHES	139
 CHAPTER SEVEN	
CONCLUSIONS AND FUTURE WORKS	140
7.1 SUMMARY OF WORK	140
7.2 CONCLUSIONS	141
7.3 FUTURE WORKS	143
 REFERENCES	143
 APPENDICES	
APPENDIX A	145
APPENDIX B	164
APPENDIX C	176

LIST OF FIGURES

FIGURE NUMBER	DESCRIPTION	PAGE NUMBER
1 1	Simplified physical system architecture for DDB	2
1 2	Distributed processing environment	4
1 3	Distributed database environment	5
1 4	A reference architecture for distributed database	7
1 5	Fragments and physical images for a global relation	8
1 6	Distributing data using manual extract.	9
1 7	Basic relation	10
1 8	Horizontal fragmentation	11
1 9	Vertical fragmentation.	11
1 10	Master/slave server process	16
1 11	X-windows client/server implementation	17
2 1	Protection of transmitted messages between database sites.	23
2 2	A reference model of distributed transaction	25
2 3	Master/slave tree.	28
2 4	Lock type compatibility matrix.	31
2 5	A local wait for graph.	33
2 6	Distributed wait-for-graph showing a distributed dead lock.	34
2 7	Local wait for graph at site 1.	35
2 8	Potential global deadlock cycle at site 1.	36
2 9	A tree of deadlock detectors.	36
3 1	Host/terminal processing mode.	46
3 2	Interaction between host and terminals.	47
3 3	Host computer in managing display of application user interaction.	47
3 4	Examples include spreadsheet, database, graphics and other applications running on PC.	48
3 5	Network processing mode.	49
3 6	Each application maintains its on data on the file server.	50
3 7	Only one user can update file X at a time.	51
3 8	Client/server processing mode.	52
3 9	Many users can update the same table at the same time.	53
3 10	A single database manager controls and maintains storage of all data in the server.	54
3 11	Client/server system.	55
3 12	Client/server system services.	56
3 13	File sharing database.	57
3 14	Client/server database.	58
3 15	General message passing.	59