

**DESIGN AND IMPLEMENTATION OF MIDDLEWARE APPLICATION FOR
SOLVING DATABASE SERVER OVERLOADING**

**An Independent Study
Presented to the
Graduate Faculty of the
Computer Studies Division
Ateneo de Davao University**

**In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Information Technology**

**By
Jaymer Mernilo Jayoma**

**ATENEO DE DAVAO UNIVERSITY
GRADUATE SCHOOL**

APRIL, 2010

Ateneo de Davao University

COMPUTER STUDIES DIVISION

GRADUATE SCHOOL

SCHOOL OF ARTS AND SCIENCES

P.O. Box 8016 Davao City

Philippines



Recommendation for Acceptance

The Graduate Faculty of the Computer Studies Division of the Ateneo de Davao University accept the Graduate Independent Study entitled

DESIGN AND IMPLEMENTATION OF MIDDLEWARE APPLICATION FOR SOLVING DATABASE SERVER OVERLOADING

which has been prepared and submitted by JAYMER M. JAYOMA in partial fulfillment of the requirements for the degree Master of Science in Information Technology.

Handwritten signature of Rey R. Aliño in black ink.

REY R. ALIÑO, M.S.

Panelist

Handwritten signature of Teresa T. Quindoy in black ink.

TERESA T. QUINDOY, M.S.

Panelist

Handwritten signature of Edwin V. Marañon in black ink.

EDWIN V. MARAÑON, M.S.

Chair Panelist

TABLE OF CONTENTS

Chapter 1	INTRODUCTION	1
	1.1 Background of the Study.....	1
	1.2 Statement of the Problem.....	2
	1.3 Objective of the Study.....	3
	1.4 Significance of the Study.....	3
	1.5 Scope and Limitation of the Study.....	4
	1.6 Definition of Terms.....	4
Chapter 2	REVIEW OF RELATED LITERATURE	6
	2.1 Distributed Data Management System Using Client-Server Technology.....	6
	2.2 Data management services of NorduGrid.....	7
	2.3 Using Component-Based Development and Web Technologies to Support a Distributed Data Management System.....	8
	2.4 A distributed Data Management System for Data Driven Application.....	10
	2.5 Review of Related Literature Summary.....	11
Chapter 3	RESEARCH DESIGN AND METHODOLOGY	11
	3.1 Conceptual Framework.....	11
	3.2 Description.....	13
	3.3 Design and Testing.....	13
	3.4 Methodology.....	14
	3.4.1 Identification and Establishment Different Database Servers.....	14
	3.4.2 Design and Implement Distributed Database Servers to Implement Load Balancing.....	14
	3.4.3 Design and Development of a middleware application that receives clients request and allocates them to available servers.....	15
	3.4.4 Create a framework for a hybrid data-synchronization.....	15
	3.4.5 Gauge the effect of the middleware on the performance of the client application.....	16
	3.4.6 Measure the effect of the middleware on the server Load.....	16
Chapter 4	THEORETICAL BACKGROUND	17
	4.1 Remote Procedure Call.....	17
	4.2 Remote Data Access.....	18
	4.3 Queued Message Processing.....	18
	4.4 Data Synchronization.....	19
	4.4.1 Two Phase Commit.....	19
	4.4.2 Replication Servers.....	20
	4.5 Theoretical Framework.....	21

Chapter 5	RESULTS AND DISCUSSIONS	22
5.1	Database Server Identification and Installation.....	24
5.2	Design and Implementation of Distributed Database.....	27
	Server	
5.2.1	Entity Relationship Diagram.....	29
5.2.2	Selected Tables.....	30
5.2.3	Stored Procedures.....	31
5.3	Middleware Design and Development	32
5.3.1	Load Balancer.....	33
5.3.2	Middleware Database Design.....	35
5.4	Hybrid Data Synchronization Framework	36
5.5	Gauging the Effect of the Middleware on the.....	46
	Performance of the Client Application	
5.6	Measuring the Effects of the Middleware on Server Load.....	52
5.7	Cost and Benefit Analysis.....	54
Chapter 6	CONCLUSIONS AND RECOMMENDATIONS	56
6.1	Conclusions.....	56
6.2	Recommendations.....	60
BIBLIOGRAPHY		61
References		61
Web Sources		61
APPENDICES		
A	List of Tables: Summary of Average Response Time of SQL.....	66
	Statement Test of Significance, and Summary of Descriptive Statistics	
	on the Running Time of the Softwares as affected by the number of	
	clients introduced in SQL Statement.	
B	Relevant Source Codes : Middleware, How to connect to a server.....	73
	Stored Procedure	
C	User Manual.....	75

ABSTRACT

As Northern Mindanao State Institute of Science and Technology (NORMISIST) expanded, it denoted more work affecting the overall performance of the database server, since it accommodated more queries thus making it overloaded. Upgrading of the ICT infrastructures were necessary to answer this problem but it was not always the viable solution especially that the organization had experienced financial constraints leaving the institute with no option but to utilize the current infrastructure. Using the existing ICT infrastructure the objective of this research was to minimize database server overloading by designing distributed systems that involved replicated database, data synchronization and measure its effects both on the client and the server. When queries were run into the existing servers (with middleware installed) the results showed better performance compared to the servers without middleware application installed. Consequently, the development of such middleware application helped the institution implement distributed systems to cater its needs and improve the overall performance.

Keywords:

Distributed-data, Distributed-Data Management System, Data-synchronization,
Client,Server

CHAPTER 1

Introduction

1.1 Background of the Study

The Northern Mindanao State Institute of Science and Technology's increasing enrolment in the past semesters post a concern with its current IT Infrastructure, especially during enrolment days where influx of enrollees were congesting in long queues. Though computerized enrolment system was in placed, a student had to spend a whole day just to enroll his or her subjects.

The enrolment process in NORMISIST was relatively slow compared to other school because of some factors affecting it. Like its ICT Infrastructure, Volume of Students, and heavy traffic of SQL statements and the results affecting the database server and the network. Those were the impending problem that the school had faced.

With this problem, one solution was to distribute the data to different sites so that clients will access the appropriate server. A distributed-data management system (DDMS) must be developed that is responsible for all data and operations on the data in a distributed computing environment. A DDMS can be viewed as a Client/Server middleware, which attempts to provide transparent access and manipulation of distributed data.

The researcher explored the Client/Server middleware specifically DDMS that resolved the current problem existing in NORMISIST. DDMS was commonly used in contemporary organizations where data resides on a variety of platforms in a variety of formats and data was distributed without or with replications. This study aimed to build

intelligence into the DDMS so it could decide which database server a client should access to minimize database server overloading.

1.2 Statement of the Problem

The present study sought to answer the following general problem: How can Database Server Overloading be minimized?

Specifically, it sought to answer the following questions:

- What database servers should be installed?
- How can a Distributed System that involves replicated database servers be designed and established?
- How can a middleware application be designed and developed to manage client request and server responses?
- What are the techniques for data-synchronization and server allocation and how can they be developed?
- What is the effect of the middleware on the performance of the client application?
- What is the effect of the middleware to the server load?

Null Hypothesis

- a. There is a significant difference in the performance of a prototype client application without-middleware and application with-middleware.
- b. There is no significant difference in the CPU usage and network traffic between servers with middleware and without-middleware.

1.3 Objectives of the Study

The general objective of this research was to establish a network of database servers and develop a middleware application to allocate server resources to different client requests.

The specific objectives were:

- To identify and establish different database servers;
- To design and implement distributed database servers to implement load balancing;
- To design and develop a middleware application that receives clients request and allocates them to available servers;
- To create a framework for a hybrid data-synchronization technique;
- To gauge the effect of the middleware on the performance of the client application;
- To measure the effect of the middleware to the server load.

1.4 Significance of the Study

Through Distributed-Data Management Systems companies can transact their processes effectively and efficiently. Recovery and back-up procedures would be easy for Database Administrators (DBA) because data was replicated across the network. In addition, new data-synchronization techniques provided the DBA the ease of managing data integrity. Network traffic would be less a problem for a Network Administrator (NetAdmin) because clients' requests were not concentrated in one location. Users would get quicker response time, thus, making them more productive and clienteles become more satisfied because of fast and reliable business transaction.

1.5 Scope and Limitation of the Study

The study generally was focused on a middleware of a Distributed-Data Management System. Intelligence was added to choose the appropriate Database Server the client applications should access.

It was implemented in the Northern Mindanao State Institute of Science and Technology (NORMISIST). A simulation of client and server interaction was used to evaluate the efficiency of the DDMS and should be limited within the scope of the specified domain.

1.6 Definition of Terms

- | | | |
|---|---|---|
| Distributed-data | - | data is distributed across several network of servers. |
| Client | - | computers who request services from a server. |
| Client/Server Technology | - | the technology of the interaction of clients and servers |
| Database | - | is a collection of logical data items in which the granularity of a logical data item maybe a file, a record, or an arbitrary collection of data fields |
| Database Management System | - | is a software package which manages the access and manipulation of a database by multiple users. |
| Data-synchronization | - | The process of synchronizing data /mirroring data across the entire distributed system. |
| Distributed-Data Management System | - | Responsible for all data and operations on the data in |

a distributed computing environment

- Middleware** - responsible for providing transparent and interactive access to resources (programs, databases) located across the network
- Queued-Message-Processing** - Messages are sent to queue in a server for sequential execution.
- Remote-data access** - Accesses databases using complex queries.
- Remote-procedure call** - Invokes a procedure from a remote site.
- Server** - Computers who grant services to a client.
- Stored Procedures** - Procedures defined in a Databases Management System who are called by clients for execution.