

# **A JDBC MIDDLEWARE FOR DATA REPLICATION**

LIBRARY  
UNIVERSITY OF MORATUWA SRI LANKA  
MORATUWA

**This dissertation was submitted to the  
Department of Computer Science and Engineering  
Of the  
University of Moratuwa  
In partial fulfilment of the requirements for the  
Degree of M.Sc. in Computer Science**



**By  
Aruna Dissanayake  
Department of Computer Science and Engineering  
University of Moratuwa, Sri Lanka  
December 2008**

**93376**

---

# Abstract

---

Data is an important part of today's IT infrastructure: both companies and state institutions rely on database systems to store most of their important data. As companies move toward systems that are distributed, research and development in data-replication is becoming increasingly important. In simple terms, replication is a form of distributing copies of data to other machines. Data replication is a widely used approach for providing high availability, fault tolerance and better performance (w.r.t reads). With the data being redundantly available from more than one source user can be shielded from network failures, delays and data source failures. One of the major goals in replication is maintaining data consistency amongst the various replicas.

However achieving consistency is a complex problem and it can adversely undermine its advantages. Therefore building an efficient, consistent replicated database is still an open research topic, though both the database community and the distributed systems community have been utilizing replication for nearly three decades.

Moreover, if two updates happen concurrently, it is often also required that each copy is updated in the same order. This is very expensive in practice and can degrade performance of the end application. Hence the choice of the consistency model mostly depends on the application using the replication scheme. For instance mission critical applications need high consistency while

mobile applications are willing to trade off consistency, for availability and performance.

Most of the researches have therefore focused on a specific application domain and interns implement a particular consistency model, which does not suit everyone. Moreover both the industry and research community seem to favour relaxed consistency models like sequential consistency, because higher consistent models are more restrictive.

In this research we propose a novel replication scheme that can be tuned to match varying degrees of consistency requirements, of a wider user community. We implement this replication scheme as a JDBC middleware. Our JDBC implementation can replicate update/insert requests across a large number of nodes (data sources/replicas). The JDBC interface hides the complexity of distribution and replication while providing a consistent and transparent view of the data store to the application developers.

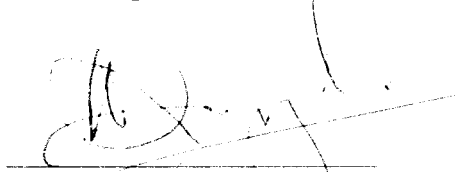
---

## Declaration of Authenticity

---

I hereby declare that the following thesis is the result of original, authentic, work by the author in which all relevant sources are properly sited and acknowledged. No sources, equipment or materials other than those mentioned have been used.

The material published here has not been submitted elsewhere with the aim of receiving credit towards a degree, or with the aim of publication prior to submitting this dissertation.



\_\_\_\_\_ 22 Dec 2008

### ***UOM Verified Signature***

\_\_\_\_\_ 22/12/08

---

# Acknowledgements

---

It is not so very important for a person to learn facts. For that he does not really need a college. He can learn them from books. The value of an education in a liberal arts college is not learning of many facts, but the training of the mind to visualize something that cannot be learned from textbooks. The Department of Computer Science and Engineering (CSE) has put this Albert Einstein saying to practice, by creating a high quality Master's program. I was privileged to be a student of its first batch. It is with deep gratitude that I mention the former Head of the Department, Dr. Sanath Jayasena, Dr Gihan Dias, and the current Head of the Department Ms. Vishaka Nanayakkara and the panel of supervisors for allowing me to proceed with this research.

I am deeply indebted to my initial supervisor Dr. Gehan Weerasinghe who taught us Distributed Computing as part of our Master's program. I got this research idea as a result of a course assignment given by Dr. Gehan to develop a data replication system, supporting two data sources. The direction and the insight he has given, during the evaluation of my assignment helped me coming up with this research idea.

After the initial breakthrough with Dr Gehan, my second supervisor Dr. A Perera gave me excellent guidance and support to complete the journey. He was instrumental in identifying the need for benchmarking the data replication scheme using a well recognized benchmarking software.

---

# Table of Contents

---

1	Introduction .....	1
1.1	Overview .....	1
1.2	Scope .....	2
1.3	Issues .....	3
1.3.1	Communication Protocol .....	3
1.3.2	Consistency Model .....	4
1.3.3	Concurrency .....	5
1.3.4	Fault Tolerance .....	6
1.3.5	Usability .....	6
2	Literature Review .....	7
2.1	Overview .....	7
2.2	History & Background .....	7
3	Methodology .....	11
3.1	Introduction .....	11
3.2	Selection of Communication Protocol .....	11
3.3	Selecting the Consistency Model .....	12
3.4	Addressing Concurrency Issues .....	12
3.5	Other Considerations .....	12
3.5.1	Error Detection and Correction .....	21
3.5.2	Usability .....	21
4	Design & Implementation .....	15
4.1	Introduction .....	15
4.2	Request Handling Flow .....	15
4.3	Data Processing Flow .....	17

4.4	Key Design Considerations.....	18
4.4.1	Communication Protocol .....	18
4.4.2	Consistency Model.....	19
4.5	About the Software .....	21
4.5.1	Server Package.....	22
4.5.2	Utils Package .....	25
4.5.3	Client Package .....	28
5	Test Result, Analysis and Discussion .....	31
5.1	Introduction.....	31
5.2	Interoperability.....	31
5.3	Benchmark Evaluations .....	32
5.3.1	TPC-App.....	33
5.3.2	Large Databases .....	36
5.4	Performance Parameter Analysis.....	38
5.4.1	Number of Replicas .....	38
5.4.2	Percentage of write operations.....	39
5.4.3	Total number of transactions .....	40
5.4.4	Network Latency.....	41
5.5	CPU & Memory Usage.....	44
5.6	Summary .....	45
6	Conclusion and Recommendations .....	47
	References.....	51
	Appendix I: Source Code.....	55



---

## List of Figures

---

Figure 1 A Typical Arrangement of Replicas and Clients .....	2
Figure 2 Request Handling Sequence .....	15
Figure 3 Data Processing Sequence .....	17
Figure 4 Hierarchy of Communication Protocol .....	18
Figure 5 Extensibility of the Communication Protocol .....	19
Figure 6 Package Structure .....	22
Figure 7 Server Package Class View .....	23
Figure 8 Server Configuration File .....	24
Figure 9 Utils Package Class View .....	25
Figure 10 Some of the Data Structure Classes .....	26
Figure 11 Message Structures .....	26
Figure 12 The Entry Data Structure .....	27
Figure 13 Class View of a Sample Client Package .....	28
Figure 14 Sample Usage of Replicated JDBC Interface .....	29
Figure 15 Bookstore Schema As Defined In The TPC-App Specification .....	33
Figure 16 Response Times for TPC-App with 5 Nodes .....	35
Figure 17 Response Times for Different Systems .....	37
Figure 18 Performance against the Number of Nodes .....	38
Figure 19 Performance vs. 'write' Operations .....	39
Figure 20 Performance vs. Number of Transactions .....	40
Figure 21 Performance vs. Network Latency .....	41
Figure 22 An Inconsistent 'update' Sequence .....	43
Figure 23 Typical CPU Graph when using the Scheme .....	44
Figure 24 CPU Graph When Simulating Many Replicas on the Same CPU .....	44
Figure 25 Memory Usage Graph .....	45



---

## List of Tables

---

Table 1 Server Configuration Parameters	24
Table 2 Entry Structure Field Descriptions	27
Table 3 Basic Configuration of Replica 1	32
Table 4 Basic Configuration of Replica 2	32
Table 5 Basic Configuration of Replica 3	32
Table 6 Expected Business Transaction Mix As Per The TPC-App Specification	34
Table 7 Altered Business Transaction Mix Used for Evaluation	34
Table 8 Initial Data Population for the Standard Test	35
Table 9 Initial Data Population for Large Database Test	36
Table 10 Consistency Model Map to network delay	42