READ-O G SYNCHRONIZATION TRANSACTION SYSTEM (ROWA-MSTS)

TITO LAMAT ANAK JAYAN

A report submitted in partial fulfilment
of the requirements for the award
of the degree of
Bachelor of Computer Science (Software Engineering)

Faculty of Computer Systems & Software Engineering
Universiti Malaysia Pahang

NOVEMBER 2009

ABSTRACT

In the world where internet does all the business nowadays, the demand of transmitted information over networks increase rapidly and the demand for steady bandwidth seems to be out of control. Particularly organizations need to provide updated data to users that might be geographically remote and handling a vast amount of requested data distributed in multiple sites. In distributed system environment, replicated data is the current trend to keep data updated in multiple sites. Replication in distributed environment has become increasingly popular due to its high degree of availability, fault tolerance and enhance the performance of a system. These advantages of replication are important because it enables organizations to provide users with access to current data anytime or anywhere even if the users are geographically remote. At certain time, access to the current data can be made anywhere in distributed systems. However, this way of data organization also introduces low data consistency among replicas when changes are made during transactions. The need to have a system to monitor this data replication arises. Read-One-Write-All Monitoring Synchronization Transaction System (ROWA-MSTS) is developed to answer the issue of problem made by the ROWA replication technique. Rapid Application Development (RAD) is the software life cycle used to develop this project due to short period of development time. This project will help to monitor the replicated data distribution over multiple sites with a better performance.

ABSTRAK

Anjakan permintaan perkhidmatan jalur lebar yang melambung dalam menguruskan kehidupan sehari-harian telah menyebabkan permintaan untuk memiliki *bandwidth* yang mantap juga telah meningkat. Organisasi khususnya,mesti memastikan setiap data berada dalam keadaan yang sentiasa dikemaskini walaupun menghadapi kekangan seperti faktor geografi yang terpinggir.Di dalam persekitaran *Distributed System* , data replikasi ialah trend terbaru untuk memastikan data dikemaskini di setiap rangkaian.Pendekatan replikasi data menjadi semakin mendapat perhatian kerana ciri-cirinya seperti darjah kehadiran data yang tinggi,mesra kesilapan dan memantapkan persembahan sesuatu sistem. Terdapat keadaan di mana data yang sama diakses serentak oleh beberapa pengakses di beberapa lokasi. Keadaan ini menyebabkan data kurang konsisten apabila terdapat kemaskini terhadap data dibuat. Maka,timbullah satu keperluan dalam organisasi untuk mempunyai satu sistem yang boleh mengawal dan menyelia data replikasi ini.Read-One-Write-All Monitoring Synchronization Transaction System (ROWA-MSTS) dibangunkan atas dasar untuk menyelesaikan kekangan yang berlaku sekiranya timbul masalah semasa data replikasi berlaku. Rapid Application Development (RAD) pula ialah pendekatan yang dipilih untuk membangunkan projek ini memandangkan cirinya yang sesuai untuk pembangunan sistem dalam masa yang singkat. Projek ini akan mengawal dan manyelia data replikasi dengan baik.

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURES

LIST OF ABBREVIATIONS

UMP	-	Universiti Malaysia Pahang
ROWA	-	Read-One-Write-All
ROWA-MSTS		Read-One-Write-All Monitoring Synchronization Transaction System

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.0 Introduction

In the world where internet does all the business nowadays, the demand of transmitted information over networks increase rapidly and the demand for steady bandwidth seems to be out of control. Particularly, organizations need to provide updated data to users that might be geographically remote while handling vast amount of requested data distributed in multiple sites.

Distributed Systems keep the internet available anywhere regardless of the geography factor through Local Area Network (LAN), Wide Area Network (WAN) and Metropolitan Area Network (MAN). Distributed Systems is a system consists of two or more computers that coordinates their processing through the exchange of synchronous and asynchronous message passing [1]. Data can be accessed through synchronous or asynchronous replication and transaction in Distributed Systems.

The need of replicated data in Distributed Systems is to ensure that any data is backed up whenever problem occurs. In addition, replicated data is important as it will increase data availability and consistency [2]. The replicated data or replicas will be copied in every computer in the Distributed Systems. There are two types of data replication namely, synchronous replication and asynchronous replication.

Asynchronous replication usually transmitted inconsistently rather than a steady stream. Asynchronous replication also caused the receiver to have problems in receiving the data from the sender. Many vendors like Lotus Notes adopted asynchronous replication as a solution for managing replicated data because asynchronous replication works reasonably well for single object updates. However, asynchronous replication fails when involving multiple objects with single update.

Therefore, synchronous replication is the answer for constraints that the asynchronous brought. Synchronous replication will guarantee data consistency since it works based on quorum to execute the operations. Plus, synchronous replication provides a *'tight consistency'* between data stores [1]. For any copy that has been updated, the updates are applied immediately to all the copies within the same transaction [1]. This will ensure that all the copies in any site are the same and consistent. A consistent copy in all sites give advantages to the organization as it provides an updated data that is accessible anytime at any place in the distributed systems environment. However, synchronous replication require vast amount of storage capacity as multiple copies of replicated data stored in many sites and expensive synchronization mechanism are needed to maintain the consistency of data when changes are made. As a result, a proper strategy is needed to manage the replicated data in Distributed Systems.

Another problem that related to this system is regarding on the monitoring issue. Legacy system is the best example. Legacy system was developed few years back. Legacy system are often considered as problematic because it runs on outdated hardware or some even hard to find nowadays [5]. There are several possibilities that there might be no proper user manual or related documents about the antique system. This problem however, can be solved since the system is developed fully in LINUX environment. Since LINUX is an open source operating system, therefore, anyone from any part of the world can modify the system parallel to the current time.

In this project, Read-One-Write-All Monitoring Synchronization Transaction System (ROWA-MSTS) will be proposed to solve the problems. ROWA-

MSTS is a system that will monitor the synchronous replicated data in distributed systems environment. A simple all-data-to-all sites replication technique called Read-One-Write-All (ROWA) technique is proposed to manage the synchronous data in Distributed Systems environment. By using ROWA, the read operation allowed to read any copies in the system. On the other hand, write operation required to write all copies of the data. Therefore, when updates of data performed, all value of data at all sites is the same. This technique provides read operations with high data availability and low communication cost. Besides that, in ROWA-MSTS system a usable interface will be developed as a result to ensure that this system is no alien to new network administrator. The system will also be able to generate the network report based on the transaction made on the particular time.

1.1 Problem Statement

Replication in distributed environment has become increasingly popular due to its high degree of availability, fault tolerance and enhance systems performance. These advantages of replication are important because it enables organizations to provide users with access to current data anytime or anywhere even if the users are geographically remote. In real working world, where data plays a crucial role like banking industry, this research is very important as the industry are dealing with consistent data like credit card numbers, amount of money transferred and other important transaction. At the same time access to the current data can be made anywhere in distributed systems. This way of data organization also introduces low data consistency among replicas when changes are made during transactions.

To understand the problem statement, consider a case of replicated data namely data x with 3 server in distributed environment. Each server is connected with one another. Define the following notations.

- PC A, PC B,PC C are the servers.
- *X is* transmitted data.
- α, β are the given locations.
- $T\alpha$, $T\beta$ are synchronous transaction at given locations.
- t is a specific time.

1.1.1 Case Study

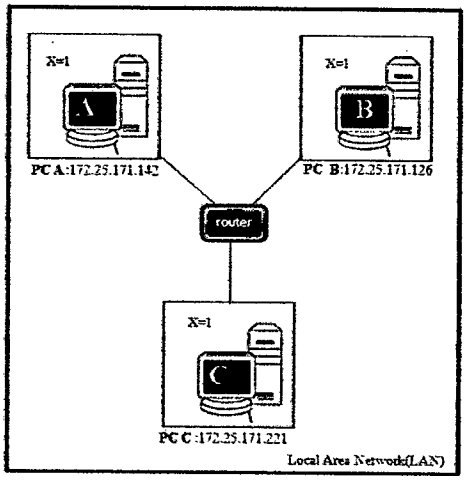

Figure 1.0 Replicated data stored in different servers

Figure 1.0 show that $X=1$ is replicated data that stored at server A, B and C at initial state.

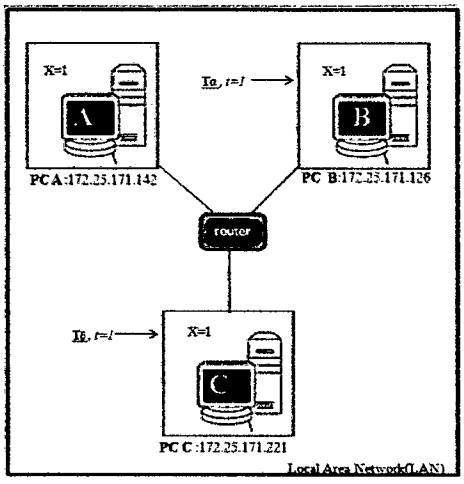

Figure 1.1 Tα request to update *X=2* at server B

Figure 2.0 shows that at time *t=1*, Tα request to update *X=2* at server B. At the same time where *t=1*, Tβ also request to update *X=3* at server C. These situation cause conflict in concurrency update requests for the particular data *X*. If there is no concurrency control for this situation, Tα will update data *X=2* at server B and Tβ will update the value of *X=3* at server C at the same time, *t=1*. At time *t=2*, *X=2* at server B, *X=3* at server C and *X=1* at server A. At time, *t=2* data is not at consistent state. An issue arises, how do we maintain a steady and consistent transaction between all replicas in different servers at one time? How do we make sure that the synchronize replicated data can be transferred correctly?

1.2 Objectives

This system is developed to achieve objectives as follow:

i. To develop a system name Read-One-Write-All Monitoring Synchronization Transaction System (ROWA-MSTS).
ii. To monitor the transaction synchronization in Distributed Systems.
iii. To apply ROWA technique in order to maintain the data consistency in distributed system.

1.3 Scope

ROWA-MSTS system is developed especially for Network Administrator that using server, fixed and wireless network, Local Area Network (LAN) connection. This system only deals with synchronous replication and the research is focus only with data consistency issue.

1.4 Thesis Organization

This thesis will be organized as follows [13]: Chapter 2 explains about research that is done regarding to this project. This chapter divided into two major parts namely, research on the existing system and about techniques and technologies that is related to this project. The research is based on the previous paper or research that had done by other scientist or any current systems that implements the techniques related to this projects. This chapter also explains about techniques or technologies relevant to this project. In Chapter 3, the approach or overall framework about the development of project are discussed. This includes techniques, methods, or approaches that is used to develop and implemented throughout the project development. Chapter 3 also explains any justification of the techniques, hardware, software and methods that is used. Chapter 4 will document all the process that is involved in the development of the projects. Contents of this are depend on types of project developed. This chapter also exhibits the source code. Chapter 5 will discuss about the findings or result that is obtained and analysis of the data. Part of this chapter including the result analysis, project constraint and suggestion for improvement. Finally, in Chapter 6 will conclude overall projects that had developed. This includes the project summary, the summary of the data that is obtained and the effectiveness of data obtained with the objectives and problem statement. This chapter also discussed about the summary of methodology and implementation that is used throughout the project. Lastly, this chapter also includes any suggestion or new proposed approach regarding on the project for the research in the future.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

This chapter will outline the general overview of any domain studies that is related to this project. This is purposely to increase the knowledge and understanding about the background of this project. This chapter also explains any research made that related to Read-One-Write-All Monitoring Transaction Synchronization (ROWA-MSTS) system regarding on the approach that is used throughout the development of this project. Chronologically, the first section is study on various replication techniques or methods that is currently used to deals with any data transaction in distributed systems environment. The second section will discuss on the already existed system that is related to this project. These existing systems can be a guide and giving some ideas on developing the project in the future. This subsection will discuss further and more specific about ROWA-MSTS system. This includes the modules involved and how the data are monitored. Next subsection explains about the software development life cycle that will be used throughout this project development along with its justification. Last section will discuss about software approach in developing the project. This includes any relevant software that might be needed.

2.1 Replication Techniques

Replication in distributed environment has become increasingly popular due to the high degree of data availability, fault tolerance and enhance the performance of a system. There are few types of replication techniques:

2.1.1 Read-One-Write-All (ROWA)

In ROWA techniques, replicas consistencies is guaranteed by the consistency of execution on one replica, but the client replicas are only updated and cannot provide accurate responses to queries. Synchronous replication methods guarantee that all replicas are maintained consistent at all times by executing each transaction locally only after all replicas have agreed on the execution order. Through this, a very strict level of consistency is maintained. However, because of the strict synchronization between replicas that is required for each transaction, synchronous replication methods have been deemed impractical and often times a centralized or client-server approach is preferred for systems that critically require strict consistency.

Any replicated database system needs to perform two main functions: execute transactions locally and make sure that the outcome of each transaction is re-erected at all replicas. The later task can be achieved in several ways. In the client-server approach, one server replica is in charge of executing/committing transactions locally before it disseminates the transactions to client replicas. This method is the closest to the single-server solution both in its performance. In ROWA-STD, the consistency of the replicas is guaranteed by the consistency of execution on one replica, but the client replicas are only lazily updated and cannot provide accurate responses to queries.

The asynchronous replication approach disseminates the transactions for parallel execution on all replicas. Each replica can answer queries and can initiate the execution of a transaction, but consistency is not always guaranteed, since connecting transactions

may be executed in different order at different replicas. These situations are detected and conflict resolution procedures are applied in order to restore consistency. If no conflicts arise, the transaction execution in the system can proceed very fast as each replica can locally execute the transactions as soon as they are received.

Synchronous replication methods guarantee that all replicas are maintained consistent at all times by executing each transaction locally only after all replicas have agreed on the execution order. This way a very strict level of consistency is maintained while providing high consistencies response to clients. However, because of the strict synchronization between replicas that is required for each transaction, synchronous replication methods have been deemed non-practical and often times a centralized or client-server approach is preferred for systems that critically require strict consistency. It is believed that the trade between performance and client response time is too unbalanced for practical needs.

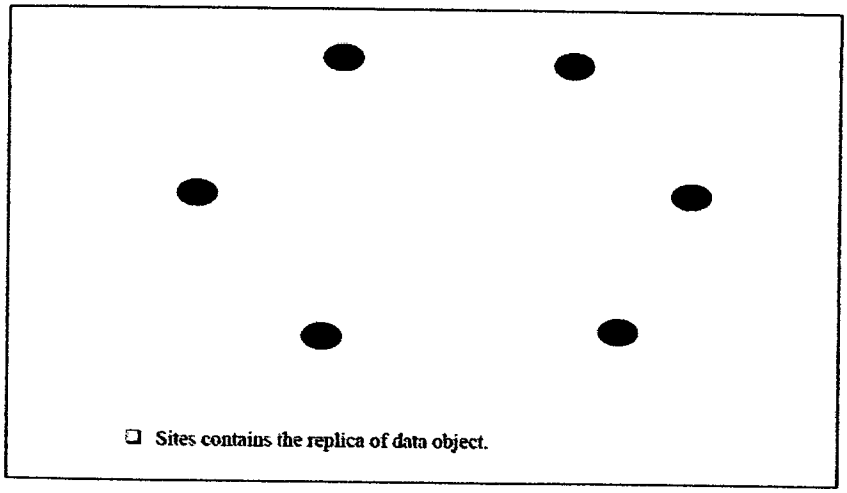

Figure 2.0: Sites contains the replica of data object

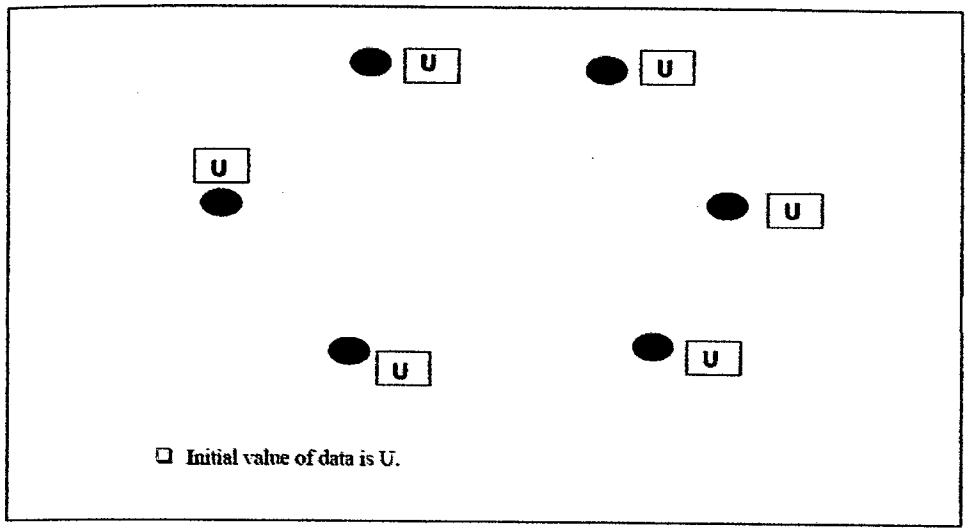

Figure 2.1 Initial value of data is U

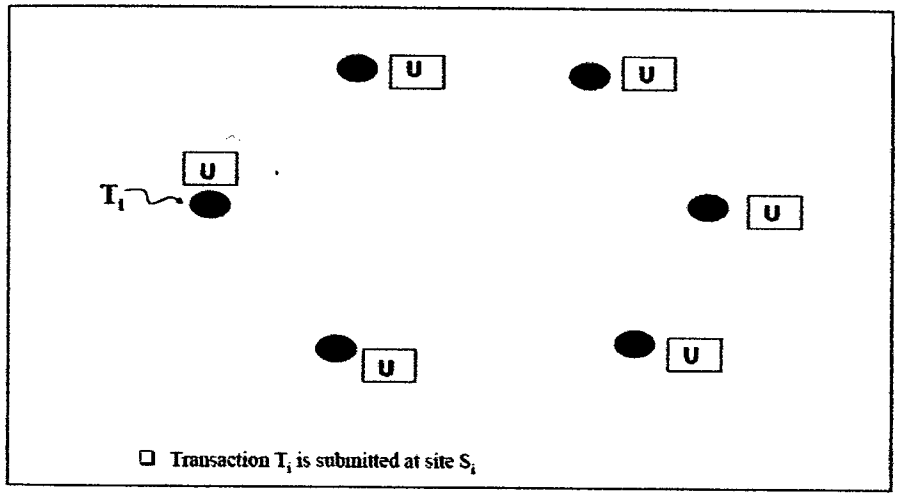

Figure 2.2 Transaction Ti, is submitted to site Si

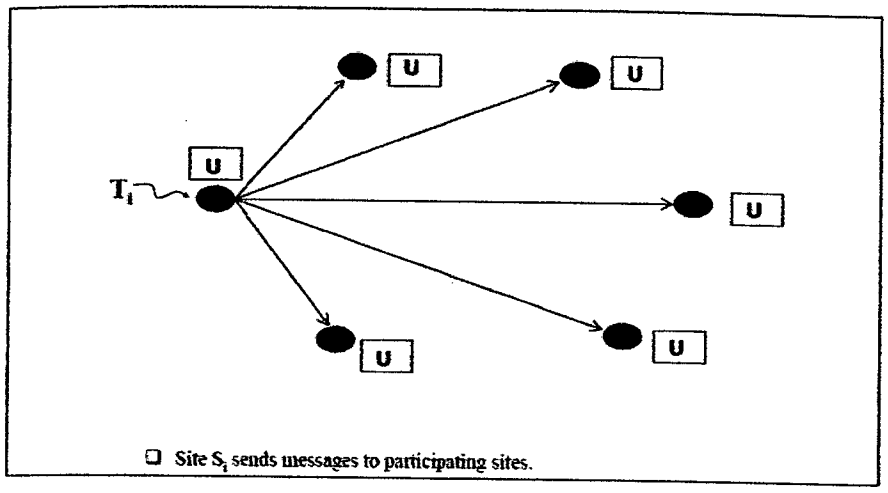

Figure 2.3 Site Si sends messages to participating sites

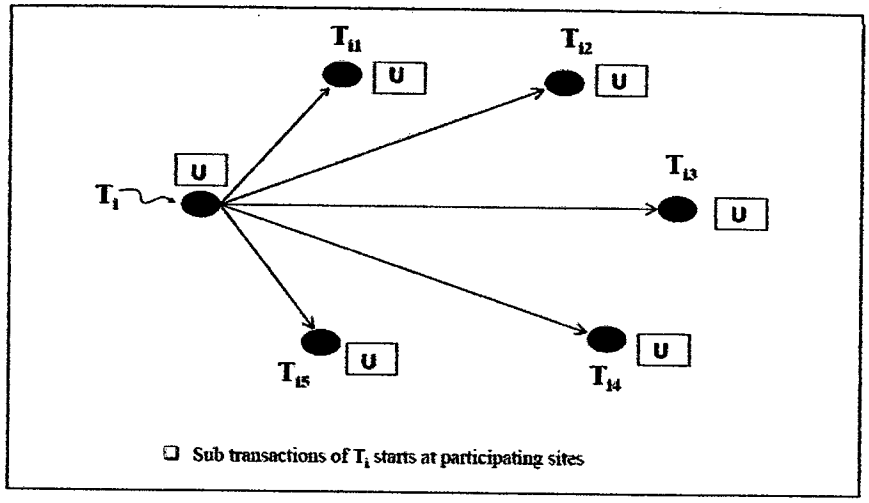

Figure 2.4 Sub transaction of Ti starts at participating sites.

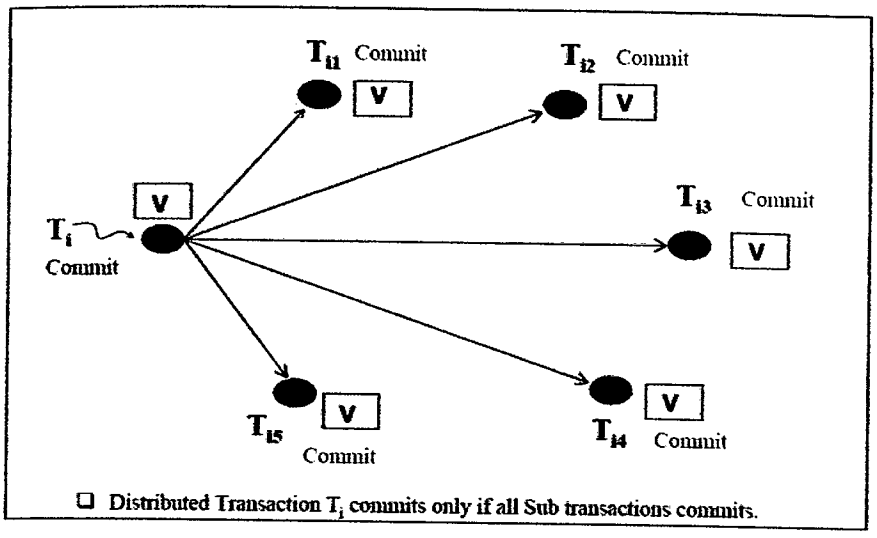

Figure 2.5 Replication Technique

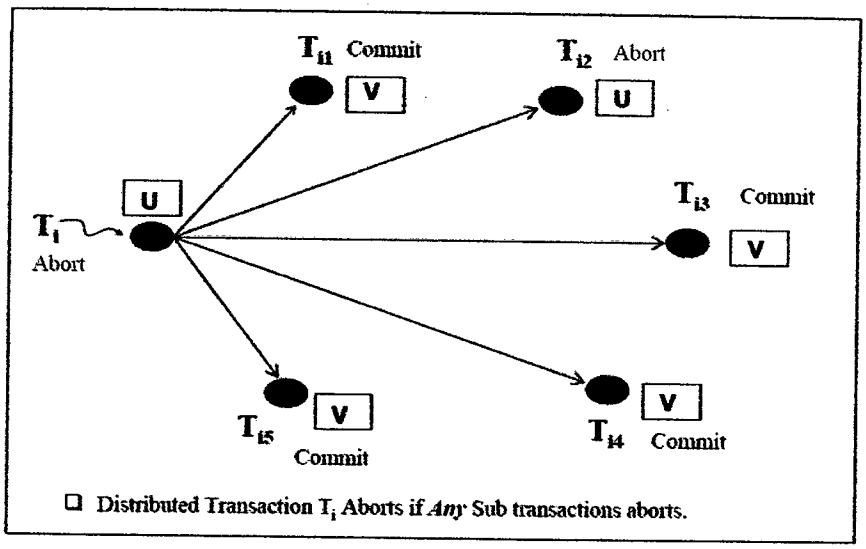

Figure 2.6 Distributed Transaction Ti aborts if any of sub transaction aborts.