Concurrency Control of a Distributed File System By using 2-Phase Locking

Su Mon Shein, Khine Moe Nwe
University of Computer Studies, Yangon
sumonshein86@gmail.com

Abstract

Nowadays, distributed system become very popular in the operational environment of computer field. Heart of every distributed system is distributed file system in which informations are separated throughout the network as files. And the server holds the location of the files in the server database. The files contain different information and can exist at any machine within the network. And all of the clients shared those informations together. When multiple clients share and make concurrent updates to the same file, requirements lead to a need to be consistency. Concurrency control problems can exist if there is no protection of concurrency control scheme. So, the concurrency control plays an important role in every distributed system. In this proposed system, concurrency control will be mainly implemented by using one of the variant of two-phase locking called two phase locking- locks write all method(2PL-LW). Balancing the stock data of a supermarket system is implemented as a case study to show the concurrency control of distributed file system by using two phase locking- locks write all method (2PL-LW).

1. Introduction

Distributed processing has become more and more important and attractive in the areas of the computer networks and distributed systems. In distributed file system (DFS), sharing of informations is in the form of files in the network. Distributed system shares all of the files in the unified view to all of the clients. A machine that holds the location of the shared file is called server [2]. Not only the files can be exists any machine of within the network but also that can exist inside the server too. And all of the machines that access that shared file from the server is called client. Two variant types of model in DFS are Remote Access Model and Upload / Download Model or Local Access Model [1]. A distributed file system may have one file server or several [2]. To guarantee that a system can run numerous processes simultaneously and correctly, one of the concurrency control technique is needed. Three alternatives of concurrency control techniques are commonly used in the distributed file system. They are concurrency control based on locking, optimistic concurrency control and concurrency control based on timestamps. Comparing these three methods for concurrency are shown below.

First, some work uses the observation that timestamp ordering is beneficial for transactions with predominantly read operations [3]. Second, locking is benefit for transactions with more writes than reads. Third, timestamp ordering aborts the transaction immediately. Fourth, locking makes the transaction wait but it can result the deadlock. As fifth, when optimistic concurrency control is used, all transactions are allowed to proceed, but some are aborted when they attempt to commit [2].

So that the optimistic concurrency control is more efficient with few conflict. But using locking method can also occur deadlock problem, (i.e. all transaction are in wait sate). But new variant of two-phase locking method called two- phase locking- lock writes all method can prevent that deadlock problem [3]. So in this proposed system, that new locking method is used to balance the stock data of a supermarket system.

2. Related Work

Several approaches are explored for controlling the concurrency of the distributed file system. From literature survey, we found many papers related with our current work. Some of them are described as follows.

In [3], have compared the two phase locking-high priority and optimistic concurrency control in multiprocessor real time databases. In that paper they compare two methods with their number of restarts, useless restarts and chained blocking. After comparing these two method, they propose two-phase locking-locks write all method which combine the advantages of two phase locking-high priority and optimistic concurrency control.

Three kinds of commonly used concurrency control techniques: concurrency control by using locking, optimistic concurrency control and concurrency control with timestamps ordering are presented in [2]. And showed the behavior of these three methods and compared their advantages and disadvantages. We can know two- phase locking is good in multiple updates and timestamps ordering is good in multiple reads.

An adaptive speculative locking (ASL) protocol is proposed in [6]. That improves performance of real-time distributed database systems by augmenting the SL protocol with four features: distributed real-time database system support; simultaneous multi-threading or page execution; control of transaction execution through transaction queue management; and restricting system memory through the use of virtual memory. The simulation results demonstrate the superiority of the ASL protocol over the SL protocols through the reduction of data contention caused by finite memory and the overall increase in transaction throughput.

In [4], distributed transactions-system for a Universal File Server has implemented. The system maintains consistency in a general purpose file-system by means of concurrency control and crash recovery. The novel distributed locking management algorithm based on the two-phase-locking (2pl) protocol is used as a main concurrency control algorithm. Their system is implemented in Flat Concurrent Prolog (FCP), a concurrent logic programming language. The features of concurrent logic languages, which are useful for implementing file and database systems, are demonstrated at that paper.

3. Theory Background

Distributed file systems support the sharing of information in the form of files throughout the network. Distributed file system is part of the distributed system that provides a user with a unified view of the files on the server. A distributed file system enables programs to store and access remote files exactly as they do local ones, allowing users to access files from any computers in the network. Many aspects of distributed file system are similar to the conventional file systems. The differences are the concepts of file service and the file server. The file service is the specification of what the file system offers to its clients. The file server is a process that runs on some machines and helps to implement the file service. A system may have one file server or several. A machine that holds the shared files locations is called the server. A machine that accesses the files is called client [2].

There are two types of model in distributed file system. They are local access model and remote access model. In local access model, entire request file is moved to requested client and all access is done by client.

But in the remote access model, the file system runs on the server, not on the client side and client perform remote open, read, write and close calls. The architecture of these two models is shown in figure 1 and figure 2.

In figure 1, the architecture of remote access model is shown. All of the file is hold by the server and if the client want to use the file, it have to make request to the server and server make all of the operation that the client request and send response back only the information to the requested client. Only the information and not the whole file. After client make the update, it send back to the server again and server make permanent update to the file. So, all of the client can’t directly use the file and server hold all of the accessomg to the file.

In figure 2, it shows the local access model or upload download model. In that model, when the client request, the server give the copy of whole file to the requested client. After client make update to the file, the updated file is send back to the server and the server replace the old file with that new file at that time [1].

By comparing these two models, local access model can cause performance decrease when uploading and downloading the file. So, in this paper, remote access model is used to improve the performance.

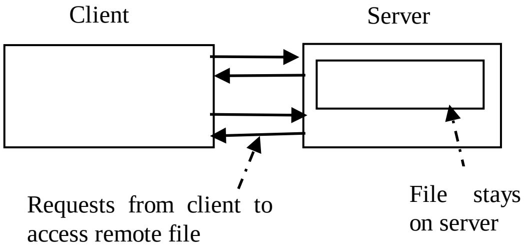


Figure 1. Remote access model

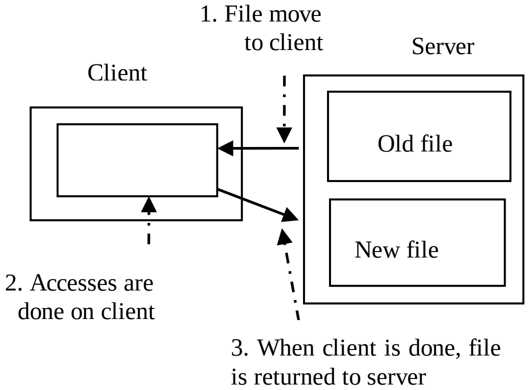

1. File move to client
Server
Client
Old file
New file
2. Accesses are done on client
3. When client is done, file is returned to server

Figure 2. Local access or upload/download model

To control the concurrency of the system, there are many kinds of two phase locking method. Among them, another method of 2PL is two phase locking-locks write all method(2PL-LW).

In 2PL-LW method, a transaction is divided into read phase and write phase. All reads precede writes. In the read phase, any update is written into a private workspace. Permanent storage or update perform in the write phase.

There are two kinds of lock modes : R lock and W lock for read lock and write lock. Lock mode will be shared lock at the start of the read phase and can acquire if there is no W lock on the desire. If the desire item is in W lock set, it waits.

In the write phase, it have to try to get W lock. To get W lock, important to be certain that there may be no other W lock on it's desire item. If any other W lock exist in the desire item, it waits. Although there is R lock on the desire item, it can get W lock. Remove all of the R locks on it's write lock set and move them to the waiting list. After W lock releases, it check are there any W lock in waiting list. If not, it check R locks exist waiting for this item. If exist, reschedule all of the R locks again appropriate with their lock mode.

So it give the highier priority on write locks and it is more suitable for the highier frequency of the updates [3].

So, two phase locking lock-writes all method is more useful in the systems with frequent update exists. By using that method, there will not be any deadlock because of giving the higher priority to writes(ie.no need to wait when read lock exists) [3].

4. Overview of the Proposed System

The architecture of this system is implemented as the architecture of distributed file system and used remote access model. In this system, the server holds all of the files locations in its database. All of the different files are seperated throughout the network and can also exist in the server too. The client have to request to the server to get information.

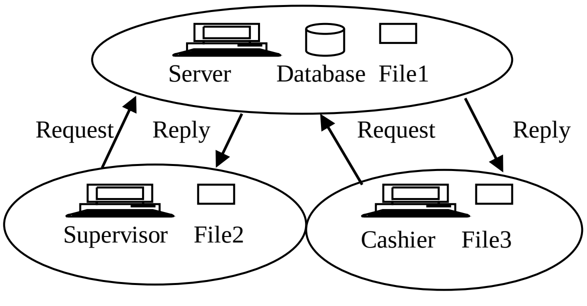


Figure 3. Overview of the system

When the sever receive the client's request, the server check it's database to forward that request to the particuler machine location whether the client request data is exist in in the server or not. If the file that client request is already exist in the server, it don't need to get from other machine.

The overview of the system is shown in figure 3.This system is implemented as the concurrency control of stock data as a case study for all cashier counter in supermarket.

5. Implementation

In this system,two types of clients are exist. They are the cashier and the supervisor. Hundred number of cashier counters for sailing are avaiable in this system.

At the supervisor portion, supervisor can update the product files when the supplier supply the products. And can also manage the cashier's data such as adding new cashier, editing cashier's profile and deleting the cashier's profile. And also the supervisor can know the products that are reach to the reorder level.

Reorder level means the limited item's quantity that the system defined. In this system, all of the users: cashier and supervisor are clients.

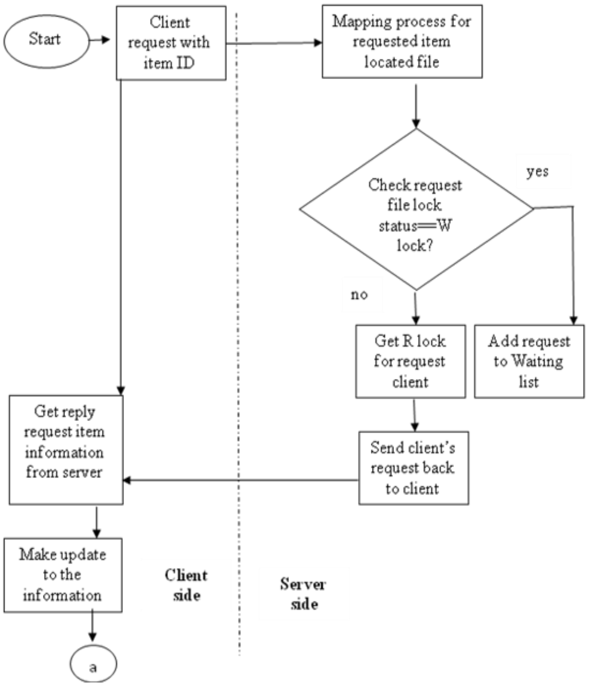
At the cashier portion, cashier can make the saling process by using the product number and product quantity of the desired product. The total amount of cost is automatically calculated by the system and can also calculate the difference of the balance between the customer paid and total cost.

Because of the applying of two-phase locking-locks write all method in this system, every transaction that the client requested are divided into two parts: read phase and write phase.

In the read phase, client request the desired information to the server. When the request arrived at the server, the server checks the lock status of the file that the client requested. If the lock status is equal to the write lock (W lock) on the desire file, incoming transaction waits. Else if the status is equal to R lock or no both read and write lock is on, it can get R lock.

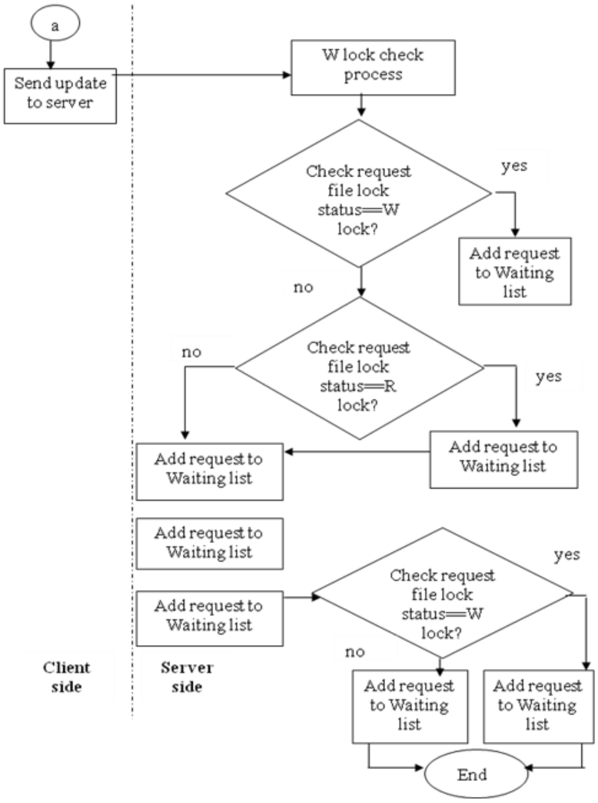
The server carries out to get the answer of client's request and send back to the requested client. The client receives the information and make update to that information.

At that time, it is started landing to the write phase. The process flow of the read phase is shown in figure 4.



Start
Client request with item ID
Mapping process for requested item located file
Check request file lock status==W lock?
yes
no
Get R lock for request client
Add request to Waiting list
Get reply request item information from server
Send client's request back to client
Make update to the information
Client side
Server side
a

Figure 4. Process flow of the read phase



a
Send update to server
W lock check process
Check request file lock status==W lock?
yes
Add request to Waiting list
no
Check request file lock status==R lock?
no
yes
Add request to Waiting list
Add request to Waiting list
Add request to Waiting list
Add request to Waiting list
Check request file lock status==W lock?
yes
no
Add request to Waiting list
Add request to Waiting list
Client side
Server side
End

Figure 5. Process flow of the write phase

The client update to the desired information and the client send updated request to the server. The server checks the requested file's lock status whether the lock status is equal to W lock or not. If the lock status is equal to W lock, that incoming transaction waits in waiting list. If not, it checks are there any R locks in its desired file. If it exists, all of the R locks in its desired file. If it exists, all of the R locks of read phase transaction on is moved to the waiting list and give W lock to the requested client update request. And then the server makes permanent update to the file. After that the server releases the W lock and checks waiting list for W lock. If exist, it gives W lock first. If waiting W lock does not exist, server reschedules the waiting R locks to that desired file. The process flow of the write phase is shown in figure 5.

In this system, all of the clients: cashier and supervisor have to make the validity checking to enter the system by using the user name, password and their post.

At the server site, controls of coming sale transaction from client site and managing of the read lock and write lock status within the system is shown in the server console. So, how the system control the concurrency of read and write phase transaction can be known.

All of the data files are separated throughout the network and their location are marked in the server database. Every time when the clients requested the file, the system has to check the file location. If the requested file is not located in the server machine, the system get the file's location from the file table on that server machine and then send this request to that machine which the file exists.

Because of the distribution of product files on the network doesn't need too much storage area and also the system can't lose all of the information at once because of the machine crash. So, the administrator doesn't need to repair all of the files and only need to repair the files that are existed in the crashed machine.

But on the other hand, the bottleneck to the server database can happen in this system.

By the usage of two-phase locking-locks write all method (2PL-LW), there will not be a deadlock because the system doesn't need to wait while the read locks exist on the desire file. That method gives high priority to the write lock. So 2PL-LW is more efficient for the applications with multiple write operations.

6. Conclusions

Nowadays, distributed systems are heavily employed in the computational field. In every distributed system, concurrency control plays an important role. In this system, concurrency problem is mainly controlled by using two-phase locking-locks writes all (2PL-LW) method and applied to the balancing of stock data on a supermarket and these stock data are distributed on branch of supermarkets.

This system improve the performance and reduce the time waste by using remote access model because there is no need to make uploading and downloading the file from server to client.

By using 2PL-LW method, there is no deadlock occur and all of the read and write transaction work serially.

References

[1] A.H. Ayman, "Advanced Topics in Distributed Systems", Computer Science Department Virginia Tech.

[2] Colouris, George; Dollimore, Jean; and Kingberg, Tim; Distributed Systems Concepts and Design, Third Edition, ISBN 0201-61918-0, 2001.

[3] A.C. Ben Kao, K.Y. Lam, "Comparing Two-Phase Locking and Optimistic Concurrency Control Protocols in Multiprocessor Real-Time Databases " TR-96-09, Department of Computer Science, University of Computer Studies, Hong Kong.

[4] E. Gudes[1], E. Reches[1], E. Shapiro[1], "distributed locking algorithm for a transactions system in flat concurrent prolog " Department of Computer Science, The Weizmann Institute of Science, Rehovot, Israel.

[5] Z. Petkovi'c, " Release Consistent Distributed File System ", Department of Computer Science College of William & Mary, zvezdan@cs.wm.edu, 16th April 2002.

[6] W. Haque and P.R. Stokes (Canada), "Adaptive Speculative Locking Protocol for Distributed Real-Time Database Systems", proceedings from Parallel and Distributed Computing and Systems, 2007.