

An Empirical Study on Extension of Three Phase Commit Protocol for Concurrency Control in Distributed System

Rukhsar Parveen*

Abstract

In distributed system two phase commit protocol and three phase commit protocol are used for concurrency control. But both protocol has disadvantage of blocking and global abortion (of transaction) respectively.

In my research paper, I have designed an algorithm for successfully submission of transaction even if any site (primary or secondary site) has voted to abort.

For this I have used a table TIT (Transaction Information Table) which has three fields namely-1) Transaction Id 2) Site Id 3) Value and three messages for communication purpose between sites. I have also used a local clock which run at regular interval for checking the value of database object (describe in detail in further part of paper).

For any transaction if any site votes to abort so it's save TIT's value="Inconsistent" with respect to that site and transaction on other sites commit successfully. So there is inconsistency that has arisen. To remove this inconsistency I have introduced the concept of local clock which runs at each site. This clock runs at regular intervals and checks for the database objects for which the TIT's value= "Inconsistent". Then it issues a transaction to remove this inconsistency. Recovery transaction will be originated by coordinator through message passing. After receiving the message inconsistent site will update its database by fetching the transaction related information from either their nearest site or the coordinator.

General Terms

Three Phase Commit Protocol, Coordinator, Consistency, Distributed System et.al.

Keywords: Concurrency Control, Two-Phase Commit Protocol, Three Phase Commit Protocol, Primary Sites, Secondary Sites.

1. Introduction

There are two types of commit protocols used for concurrency Control. One is the two phase commit protocol and other is the three phase commit protocol. In two phase commit protocol the sites having more queries become primary site and those which are having less queries become secondary sites.

1.1. Two Phase Commit Protocol

There are two phases in two phase commit protocol:

1. Voting Phase: In this the primary site ask all secondary sites to vote either to commit or to abort .Then secondary sites cast their votes.
2. Commit Phase: Based on the votes cast by secondary sites, coordinator decides to commit if all secondary sites votes commit or abort if any of the secondary sites votes to abort and after making decision coordinator notifies the result to all the sites.

* Invertis Institute of Engineering and Management, Bareilly, Uttar Pradesh, India. E-mail: rprukhsar@gmail.com

Two phase commit protocol has a blocking disadvantage in which either the coordinator or some participating site is blocked but still 2 PC is most widely used in distributed systems.

1.2. Three Phase Commit Protocol

Three phase commit protocol was introduced as a remedy to the blocking disadvantage of two phase commit protocol. It is the extension of two phase commit protocol. It introduces an extra phase which ensures the non blocking property of this protocol. The site on which transaction is generated becomes coordinator and other becomes cohorts. The three phases are as follows:

1. Voting Phase: Firstly the site at which transaction originates become coordinator and then it asks the other cohorts to vote to either commit or to abort. The cohorts cast their votes to coordinator and based on the voting done by cohorts, coordinator decides to commits the transaction if all cohorts are in favour of commits. Otherwise it decides to abort even if any of the cohorts is against of committing the transaction.
2. Prepare to Commit: Now in this phase the coordinator notifies it's decision to all cohorts. If his decision is to committing the transaction then a message "enter into ready to commit stage" is send to all cohorts.
3. Decision Phase: If the decision made by coordinator is to commit the transaction then it will send global_commit to all cohorts and wait for receiving their acknowledgement. And after receiving their acknowledgement it decides to commit the transaction. If the decision made by coordinator is to abort the transaction then it will send global_abort to all sites and aborts the transaction. In this protocol the final decision is made after receiving the acknowledgements.

2. Proposed Work-Extension of Three Phase Commit Protocol

Three Phase Commit Protocol solves the problem of blocking disadvantage of two phase commit protocol but do not show good result when any SITE votes to aborts because Its abort whole transaction. By my algorithm this global abortion problem has been solved in following way:-

The site at which the transaction originated becomes the coordinator and other becomes the cohorts. I have introduced new table called TIT (Transaction Information Table) which has three fields namely-

1. Transaction ID:- In our protocol every transaction has a unique no for its (transaction) identity.
2. Site ID:- This field has unique no. for each site for its (site) identity.
3. Value:- This field has two values for each site either "Consistent" (if the transaction commit at that site) or "Inconsistent" (if the transaction abort at that site).

I have also used 3 messages-

1. Message 1- Coordinator send message to inconsistent site asking the site to complete the transaction. It has transaction id and id of inconsistent site along with the message.
2. Message 2- Inconsistent site send this message to nearest site for updating information. It will send transaction id and site id along with message.
3. Message 3- Inconsistent site sends this message to coordinator after completing the transaction with transaction id and site id.

(For Phase 1 Phase 2 Phase 3 refer to three phase commit protocol phases which have already described in paper part-2 Introduction)

2.1. Algorithm

If (all the sites vote to commit)

Begin

Then start Phase two and send "Enter into ready to Commit stage" message to all primary sites and enter Into Phase Three sending global commit to all Sites and upon receiving the acknowledgement Committing the transaction.

End

Else

{

If (any one them votes to abort)

Begin

Go to phase 3 and send global commit to all sites who have voted to commit and set the value in table TIT="INCONSISTENT" with respect to that site who have abort.

End

Now at regular interval when local clock run recover algorithm start:

Check status of the database object which will be in use.
If (VALUE=INCONSISTENT)

Begin

Coordinator checks the site id and sends message1 to that inconsistent site.

End

When inconsistent site receives the message1 (Transaction id, Site id).

It sends the message2 to nearest primary/secondary site.

If (site is not idle)

{

Inconsistent site sends the message2 at regular interval for a

Time which is fixed.

If (site becomes idle)

Begin

Step 1-It check the transaction id and site id and all sent all transaction related information to that inconsistent site

Step 2- After receiving the information inconsistent site update database according to it and set TIT's value="consistent".

Step 3-send message3 to coordinator site and coordinator will set the TIT's value="consistent".

Now coordinator site will check the TIT's value field

If (value! ="inconsistent")

{

Delete the table related to that transaction

}

End

Else

Begin

It sends the message2 to coordinator

If (coordinator is idle)

Begin

Repeat step 1, 2 and 3.

Now coordinator site will check the TIT's value field

If (value! ="inconsistent")

{

Delete the table related to that transaction

}

End

Else

{

Inconsistent site will send the message2 at regular interval until all the information will not received.

}

}

2.2. Analysis of Extension of Three Phase Commit Protocol

This Extension of three phase commit protocol, like the original 3 phase commit protocol avoids the blocking limitation of 2 phase commit protocol and global abortion problem of 3 phases commit protocol. It also removes the global abortion problem caused by primary sites.

As i have seen in this protocol that when a transaction is issued and any sites votes to abort then coordinator set the TIT 's value ="inconsistent" with respect to that site.

So there is inconsistency that has arisen. To remove this inconsistency I have introduced the concept of local clock which runs at each site. This clock runs at regular intervals and checks for the database objects for which the TIT's value is set as "Inconsistent" Then it issues a transaction to remove this consistency. Thus in regular intervals the

local clock runs at each site to remove inconsistency. Recovery transaction will be originated by coordinator through message passing. After receiving the message inconsistent site will update its database by fetching the transaction related information from either their nearest site or the coordinator. After recovery transaction coordinator site delete all the information related to that transaction in the TIT table.

2.3. Advantages of Extension of Three Phase Commit Protocol

Extension of 3 phase commit protocol proves to be very useful in the cases where I perform some transaction on single database object in distributed systems. This protocol ensures the commitment of transaction originated at a primary site even when some important site has voted to abort. This is different from the original 3 phase commit protocol in which if one of the primary site or secondary site votes to abort then the whole transaction will be aborted. So this proposed protocol ensures that if the transaction has originated from a primary site and even if any primary sites/secondary sites have voted to abort then also the transaction will be commit

Second advantage of this protocol is that this protocol is non blocking protocol. If the coordinator fails then one of the primary sites is chosen as the new coordinator and the transaction Proceeds.

Third advantage of this protocol is that after completing the transaction coordinator will delete the unused information from the TIT table that will save the memory storage.

3. Conclusion and Future Work

It can be concluded that the Extension of three phase commit protocol can be used only when there is a transaction that accesses a single database object and its ensures commitment of transactions in all condition (when any primary or secondary site vote to abort). As it

is showing good results in all the failure of a site so this will become very powerfull algorithm in the field of Bank system, Industries, Multinational companies and at all places where data is globally distributed and updated by different sites. It's ensuring that value you are accessing is globally correct.

It's also save time as in three phase commit protocol we have to wait or restart the transaction if any of site vote to abort. But by using messages and TIT table all data can be easily fatch from near Primary/Secondary site or from coordinator after the commitment of transaction on other site.

Algorithm condition of sending message by inconsitent site at the regular interval to nearest consistant primary/secondary or coordinator site for latest value is playing a good role in successfull upadation of database.

After successfully run the transaction at all sites a checking is occur in TIT table for the column of TIT's value and if it's find all value is consistant for all site so its truncate the table and use that space of memory for new transaction.

So it definitely reduces the probability of a transaction abortion and improves the overall performance of distributed systems.

In future I will try to eliminate number of the messages to reduce the overheads.

References

1. Tanenbaum, A. S. (2006). *Distributed Systems: Principles and Paradigms* (2nd Ed.).
2. Silberchatz, A., Korth, H. F. & Sudarshan, S. (1983). *Database System Concepts- Fundamentals of Database* (4thEd.). Tata McGraw Hills.
3. http://en.wikipedia.org/wiki/Threephase_commit_protocol
4. Power Point presentation of IIT Kharagpur. Retrieved from http://www.facweb.iitkgp.ernet.in/~pallab/dist_sys/Lec-11-CommitProtocols.pdf