Suggested Cluster Co-ordinator Protocol for Concurrency Control in Distributed DBMS

Dr. Deepali Sawai
Institute of Industrial & Computer Management & Research, (IICMR), SP Pune University
Sector 27 A , Nigdi Pradhikaran, Pune, Maharashtra , India
deepalisawai@yahoo.com

Abstract—Distributed Database Management system is a need of the hour. The concurrent transactions are not only important in any Data Base Management System but are also important in Distributed DBMS. For the data to be in consistent state in the database, various concurrency control protocols are developed and are in use. Every protocol has it's own advantagesand limitations, depending upon the application, type of data and the way it is distributed, the protocols are used. The following paper suggests a protocol : "Cluster Co-ordinator protocol " for concurrency control. The protocol uses a co-ordinator for the group or cluster of sites responsible for locking of the data in that cluster. The locking requests would be mainly between co-ordinators thus reducing the data traffic.

Keywords—DBMS, DDBMS, Lock based protocols, Concurrency control, transactions, ACID properties.

I. INTRODUCTION

A Distributed database is a database that is under the control of a central DBMS in which storage devices are not all attached to a common CPU. It may be stored in multiple computers located in the same physical location or may be dispersed over a network of interconnected computers. Distributed Database (DDB)can be defined as a collection of multiple logically interrelated database distributed over a computer network and a Distributed Database Management System (DDBMS) as a software system that manages a distributed database while making the distribution transparent to the user.[1].

A distributed database system consists of loosely coupled sites that share no physical component. The database systems that run on each site are independent of each other and the transactions may access data at one or more sites. Concurrent transactions :Any transaction executing in the DBMS has to have 4 properties :

i. It should be Atomic
ii. If the data before the execution of the transaction is in consistent state , it should remain in consistent state after the execution.
iii. Even if multiple transactions are running concurrently in the DBMS they should work in isolation with each other.
iv. The data should be durable, or should remain permanent, even after the transaction is complete.

These are popularly known as ACID properties of a transaction.

The DBMS has Concurrency control protocols which need to be followed by all the transactions running concurrently so that the data remain consistent in the database. When the DBMS is Distributed DBMS, the concurrency control becomes still more complicated since the transactions which are running on various different sites run concurrently and may need the data which is distributed over various sites . The DBMS provides two types of protocols in such cases

i. Lock based protocol
ii. Time stamp protocol

There are various protocols developed under these two categories of protocols.

II. LITERATURE REVIEW

According to Ramez Elmasri, Shamkant B Navathe[2], the DDB technology emerged as a merger of two technologies :database technology and network and data communication technology. Distributed databases and Client-server architectures in the development of database technology are closely tied to advances in communication and network technology

Ming Xiong, Krithi Ramamritham, Jayant R. Haritsa and John A. Stankovic [3], in their white paper "MIRROR: A State-Conscious Concurrency Control Protocol for Replicated Real-Time Databases" state that *Data replication* can help database systems meet the stringent temporal constraints of current real-time applications ,

a pre-requisite for realizingthe benefits of replication, however, is the development of high-performance *concurrency control* mechanisms.

Mohamed M. Saad and Binoy Ravindran[4] have Presented HyFlow — a distributed software transactional memory(D-STM) framework for distributed concurrency control. According to them Lock based concurrency control suffers from drawbacks including deadlocks, livelocks, and scalability and composability challenges. These problems are exacerbated in distributed systems due to their distributed versions which are more complex to cope with e.g., distributeddeadlocks. Accroding to them D-STM are promising alternatives to distributed lock-based concurrency control for distributed systems

Further according to Raheel M. Hashmi & Hashim Ali[5] ,the present structure of Distributed locking manager in market is vendor and hence protocol specific. However, the generic logic used in each of these solutions maps to the same levels of functionality. A slight view at the horizon suggests that the growth of distributed networks is not going to slow down in near future. Distributed locking Manager has not reached a standard implementation and needs to be standardized

According to Abraham Silbetschatz, Henry F Korth , S Sudarshan[6] , a compromise between the advantages and disadvantages in concurrency control in Distributed database systems can be achieved through distributed – lock manager approach, in which the lock-manager function is distributed over several sites.

III. DISTRIBUTED DATABASE MANAGEMENT SYSTEM(DDBMS)

In Distributed Database Management System, the main focus is distributing the data on various sites so that the data is available all the time. The data is distributed and stored in various ways.

1. Distributed Data Storage

The distributed data storage uses relational data model. The data is stored in any of the three ways:

Data storage by Replication

Here the system maintains multiple copies of data, stored in different sites, for faster retrieval and fault tolerance. Replication is nothing but to store several copies of a relation or relation fragment. A relation or fragment of a relation is replicated if it is stored redundantly in two or more sites. Full replication of a relation is the case where the relation is stored at all sites. Fully redundant databases are those in which every site contains a copy of the entire database.

There are advantages as well as disadvantages of replications. The first important advantages is the availability - failure of site containing a relation does not result in unavailability of the required data since replicas of the relation exist. Also queries on the relation may be processed by several nodes in parallel. Replication also reduces the data transfer since the data is available locally at each site.

The disadvantages include increased cost of updation, since each replica of the relation must be updated. The concurrent updates to distinct replicas may lead to inconsistent data unless special concurrency control mechanisms are implemented, thus increasing complexity of concurrency control.

Data storage by Fragmentation

Here the relation is partitioned into several fragments stored in distinct sites. Fragmentation consists of breaking a relation into smaller relations or fragments and storing the fragments possibly at different sites. Division of relation into fragments which contain sufficient information to reconstruct the relation under consideration. When a relation is fragmented, one must able to recover the original relation from the fragments.

a. Horizontal fragmentation:

In this each tuple of relation is assigned to one or more fragments. i.e. each fragment consists of a subset of rows of the original relation. The union of the horizontal fragments must be equal to the original relation.

b. Vertical fragmentation:

In this the schema for relation is split into several smaller schemas i.e. each fragment consists of a subset of a columns of original relation.

All schemas must contain a common candidate key (or super key) to ensure lossless join property.

A special attribute, the tuple-id attribute may be added to each vertical fragment to serve as a candidate key.

Data storage by Replication and Fragmentation in combination

Here the relation is partitioned into several fragments: system maintains several identical replicas of each such fragment.

2. Distributed Transactions

In distributed environment, transaction may access data at several sites. Each site contains 2 sub-systems : a local transaction manager and a transaction coordinator. Transaction Manager is responsible for:

a. Maintaining a log for recovery purposes

b. Participating in coordinating the concurrent execution of the transactions executing at that site.

Each site has a transaction coordinator, which is responsible for:

a. Starting the execution of transactions that originate at the site.

b.Distributing subtransactions at appropriate sites for execution.

C.Coordinating the termination of each transaction that originates at the site,which may result in the transaction being committed at all sites or aborted at all sites.

3. Concurrency Control

The number of transactions executing parallel needs many issues to be tounched upon. Though the number of transaction which can execute concurrently can be controlled by the user. As in Netezza datawarehouse architecture [2]

The concurrency control schemes are modified for use in distributed environment.

a. Time stamping

Timestamp based concurrency-control protocols can be used in distributed systems. Each transaction must be given a unique timestamp. To generate a timestamp in a distributed fashioneach site generates a unique local timestamp using either a logical counter or the local clock.Global unique timestamp is obtained by concatenating the unique local timestamp with the unique identifier.

b. Lock based Approaches

i. Single-lock Manager Approach

In this protocol, the system maintains a single lock manager residing on chosen site, say S_i. When a transaction needs to lock a data item, it sends a lock request to S_i and lock manager determines whether the lock can be granted immediately. If yes, lock manager sends a message to the site which initiated the request. If no, request is delayed until it can be granted, at which time a message is sent to the initiating site. The transaction can read the data item from *any* one of the sites at which a replica of the data item resides. Writes must be performed on all replicas of a data item. Advantages of this scheme is its implementation as well as deadlock handling is Simple. The disadvantage are, the lock manager site becomes a bottleneck also the system is vulnerable to lock manager site failure.

ii. Distributed Lock Manager

In this approach, functionality of locking is implemented by lock managers at each site. Lock managers control access to local data items using special protocols for replicas. The advantage of this approach is the work is distributed and can be made robust to failures, but the disadvantage is the deadlock detection is more complicated.

There are several variants of this approach

1. Primary copy
2. Majority protocol
3. Biased protocol
4. Quorum consensus

1. Primary Copy

In this approach one replica of data item is chosen to be the primary copy. Site containing the replica is called the primary site for that data item. Different data items can have different primary sites. When a transaction needs to lock a data item *Q*, it requests a lock at the primary site of *Q*.Implicitly gets lock on all replicas of the data item. The main advantage of this approach is its simple for implementation. But the disadvantage is, if the primary site of Q fails, *Q* is inaccessible even though other sites containing a replica may be accessible.

2. Majority Protocol

In this approach, Local lock manager at each site administers lock and unlock requests for data items stored at that site. When a transaction wishes to lock an unreplicated data item *Q* residing at site S1, a message is sent to S1's lock manager. If *Q* is locked in an incompatible mode, then the request is delayed until it can be granted. When its granted, the lock manager sends a message back to the initiator indicating that the lock request has been granted.

In case of replicated data, If *Q* is replicated at n sites, then a lock request message must be sent to more than half of the n sites in which *Q* is stored. The transaction does not operate on *Q* until it has obtained a lock on a majority of the replicas of *Q*.When writing the data item, transaction performs writes on all replicas. The advantage of this approach is it can be used even when some sites are unavailable. The disadvantage is it requires $2(n/2 + 1)$ messages for handling lock requests, and $(n/2 + 1)$ messages for handling unlock requests. Also there is potential for deadlock even with a single item.

3. Biased Protocol

In this protocol, there is a local lock manager at each site. When a transaction needs to lock data item *Q in* shared mode, it simply requests a lock on *Q* from the lock manager at one site containing a replica of *Q*. When transaction needs to lock data item *Q in e*xclusive mode, it requests a lock on *Q* from the lock manager at all sites containing a replica of *Q*. The main advantage is it imposes less overhead on read operations but the disadvantage is additional overhead is required on writes.

4. *Quorum Consensus Protocol*

This protocol is a generalization of both majority and biased protocols. Each site is assigned a weight. Two values : Qr – read quorum & Qw – write quorum for data item x are decided / chosen using formulae., if S is the total weights of all sites at which data item x resides, it works on two values Q_r and Q_w . Such that $Q_r + Q_w > S$ and $2 * Q_w > S$. Quorums can be chosen (and S is computed) separately for each item . To execute a read operation, enough replicas of item must be locked so that the total weight is >= Q_r And to execute write operation the sum of the site weights is >= Q_w The advantage of this approach is it permit the cost of read/write reduced by appropriately defining read & write quorums.

IV. NEED OF ADDITIONAL CONCURRENCY CONTROL PROTOCOL

As it can be seen from the various approaches for concurrency control in distributed system, there are advantages and disadvantages for all the approaches. The applications can decide which protocol need to be applied,depending upon the requirement.

If all the advantage of the approaches mentioned above are clubbed together, and if a protocol is formulated, for some specific application it would be useful.

V. SUGGESTED PROTOCOL

The study proposes the following protocol for concurrency control.

A. *Suggested protocol*

The protocol which is suggested is a lock based protocol. The regular lock based protocol works on the shared and exclusive lock modes, when a request of lock on a data item is given by any transaction, the lock manager decides whether the data item can be given to the transaction by locking it. When the data is distributed, in DDBMS, the site where the transaction is initiated consulting the catalogue asks for the required data item to the site where the data item is stored depending upon the various protocols mentioned in the section II above.

To save the time, in the proposed protocol, the sites are grouped into cluster having two or more sites in a cluster. The following figure Figure 1 depicts how the setup is done for the Distributed Database System

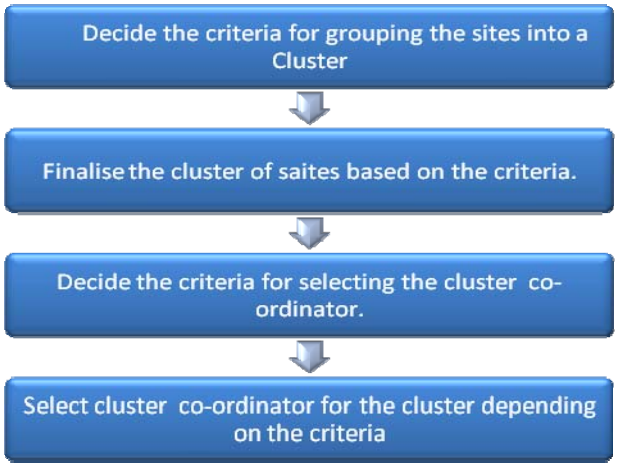

Figure 1: Setting up the System

The Clusters of sites can be decided based on criteria like :

- Location of the site :Physical distance between the sites
- Data distribution & availability of the data: The way the data is distributed with either horizontal or vertical partition, and its availability according to the application. Like the replicated data can be grouped into single cluster or other ways depending upon the application.

Once the clusters are formed, the cluster co-ordinator for the cluster would be decided. The cluster co-ordinator can be decided depending on the criteria like

- Configuration: The site with higher configuration, higher processing capability rather that storage capability should be selected as the cluster co-ordinator. More members in the cluster more messages need to be sent to and fro for the lock request,
- In case of failure of the Cluster co-ordinator site, a new Cluster co-ordinator need to be appointed considering the criteria mentioned above to select the co-ordinator. The failure of the site can be understood if there is no response from any member of the cluster under consideration for the specified time limit. A request would be raised by the sender, site where the transaction was initiated and needing the data item from the Cluster.
- Each Cluster co-ordinator maintains a catalogue for the members of the cluster. This catalogue has the information of the data items along with the lock information like locking mode, locked by which co-orditator site.

The following figure-2 shown below depicts the formation of the Cluster and selection of the Cluster co-ordinator as site number 10 from the cluster of sites number 1, 46, 82, 8, 19, 100 for the Cluster. Here the sites are given numbers instead of the location name just for the convenience.

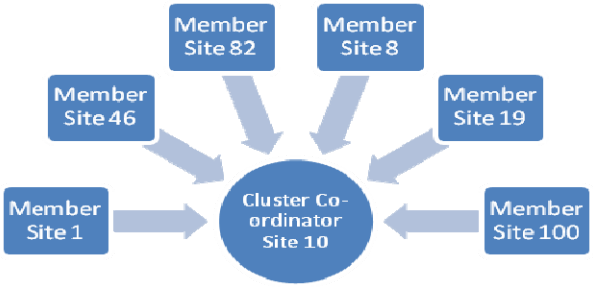

Figure 2 :Site 10 chosen as the Cluster co-ordinator for the Cluster based protocol

Flow of Working :When the transaction is initiated at one of the site and it requires to acquire a lock in required mode (Shared / Exclusive) for the dataitem at other sites, the requestis first sent to it's Cluster co-ordinator where the site under consideration is the member. The Cluster co-ordinator then checks the compatibility mode for locking the data item which may be present in the same Cluster, if yes it locks them updates the catalogue and sends the request to all Cluster co-ordinators having the data item in the member site.The Cluster co-ordinators after receiving the request, checks the compatibility of the lock for the data item in its catalogue. If it is in compatibility mode then the lock is acquired and the "Acquired" message is sent to the Cluster co-ordinator which has sent the request but if not in compatibility mode, "No compatibility" message is sent to the Cluster co-ordinator of the initiator site. If "Acquired" message is received from total n/2+1 Cluster co-ordinators, 'n' being the total number of Cluster co-ordinators, then the transaction continue it's execution else it waits for predefined time period and then aborts.

VI. CONCLUSION

The suggested protocol would be more useful in the following scenario:

i. When the number of sited participating in the Distributed Database system is large.
ii. Using the protocols which are already present have their own limitation as discussed, if the suggested protocol is used the site to site communication would be reduced since the communication would be only between the Cluster co-ordinator.
iii. Depending upon the requirement of the application, the Clusters can be changed with respect to the criteria.
iv. Also the co-ordinator can be changed is required.

Though there are advantages of this protocol, There are some limitations too,

i. If the cluster is too small like only 2 sites in a cluster, the communication may not be decreased much.
ii. If the Cluster c0-ordinator is down the transactions may not able to proceed till the other cluster co-ordinator is not appointed.

REFERENCES

[1] Ramez Elmasri , Shamkant B Navathe, Fundamentals of database system , 5th edition, P.876
[2] Ramez Elmasri, Shamkant B Navathe, Fundamentals of Database Sustems , 5th edition
[3] Ming Xiong, Krithi Ramamritham, Jayant R. Haritsa and John A. Stankovic – "MIRROR: A State-Conscious Concurrency Control Protocol for Replicated Real-Time Databases"
[4] Mohamed M. Saad, Binoy Ravindran, " Supporting STM in Distributed Systems: Mechanisms and a Java Framework"
[5] Raheel M. Hashmi & Hashim Ali ," Distributed Locking Manager (DLM) - The evolution of concurrency control in distributed networks"
[6] Abraham Silbetschatz, Henry F Korth , S Sudarshan , Dtabase system concepts, 5th edition
[7] winter corporation white paper " concurrency & Workload Management In netezz