# Transaction processing in Distributed Database Systems





## Two-phase commit protocol (2PC)

To enforce atomicity of distributed database transactions a global recovery manager (coordinator) maintais information needed for recovery

Global COMMIT or global ROLLBACK is performed in two phases

PHASE 1

All participating systems inform a coordinator that a transaction at a local system is completed

A coordinator sends a message "can commit ?" to local systems

All participating systems force-write all log records and information needed for recovery and send "ready to commit" message to a coordinator





## Two-phase commit protocol

PHASE 1 (continuation)

If a participating system cannot force-write all log records then it sends "cannot commit" message to a coordinator

PHASE 2

If all participating systems reply with "ready to commit" message then a coordinator sends "commit" message to all participating systems

Each participating systems complete the transactions by writing `COMMIT` to a transaction log and optionally permanently updating a database

If at least one of participating systems reply with "cannot commit" message then a coordinator sends "rollback" message to all participating systems





## Problems

2PC protocol is a blocking protocol

Blocking protocol means that if a coordinator fails then all participating sites must wait until a coordinator recovers

If a coordinator and one of participating transactions fails together then the distributed transaction becomes nondeterministic

It is impossible to ensure that all participants got "commit" message in the second phase

Then some of participants may commit independently on the other participants





## Three-phase commit protocol (3PC)

In 3PC the first phase is the same as in 2PC

The second phase is divided into PREPARE-TO-COMMIT and COMMIT phases

PHASE 1

All participating systems inform a coordinator that a transaction at a local system is completed

A coordinator sends a message "can commit ?" to local systems

All participating systems send "yes" message to a coordinator

If a participating system send a message "no" ten coordinator sends "abort" message





## Three-phase commit protocol (3PC)

PHASE 2

If all participating systems reply with "yes" message then a coordinator sends "pre commit" message to all participating systems and waits for "acknowledgement" message

Each participating system replies with "acknowledgement" that it is ready to commit;

At this point each participating system is aware that global commit is possible

If a participating system is not able to reply with "acknowledgement" message the transaction is aborted by a coordinator

## Three-phase commit protocol (3PC)

PHASE 3

A coordinator sends "do commit" message to all participating systems

Each participating system replies with "has committed" message after `COMMIT` operation was successful





## Problems

Dealing with multiple copies of the data items

Failure of individual sites

Failure of communication links

Distributed commit (2PC,3PC)

Distributed deadlock





## Solutions

A particular copy of a data item is designated as a distinguished copy

Extended centralized locking is used to control concurrency

The following methods are based on an idea of distinguished copy:

-primary site technique

-primary site with a backup site

-primary copy technique

Distributed concurrency control based on voting





## Primary site technique

A single primary site becomes a coordinator site for all data items

All locks are kept at that site and all lock/unlock requests are handled there

If all transactions follow 2PL protocol then conflict serializability is enforced

Information about all locks is kept at a primary site and data items can be accessed at the remote sites

When a data item is updated at a remote site all its copies must be updated before a write lock is released

Locking performed in one primary site overloads that site and becomes a bottleneck

Failure of a primary site blocks entire system





## Primary site with backup site

A single primary site becomes a coordinator site for all data items

The second site is designated as a backup site

If all transactions follow 2PL protocol then conflict serializability is enforced

Information about all locks is kept at a primary site and and at a backup site and data items can be accessed at the remote sites

When a data item is updated at a remote site all its copies must be updated before a write lock is released

Locking performed in primary site and in a backup site slows down a process of acquiring a lock

Failure of a primary site does not block entire system





## Primary site with backup site

A process of recovery from failure of a primary site is simpler and faster

Locking overloads both primary and backup sites

01 Transaction processing in DDBS

## Primary copy technique

Many sites become lock coordinators for all data items by having distinguished data items stored at different sites

Failure of one site affects only transactions that use locks on items whose primary copies are located at the site; the other transactions are not affected

This technique can also use backup sites to improve reliability and availability



01 Transaction processing in DDBS

## Voting based techniques

In voting technique there is no distinguished copy

Lock request is sent to all sites that have a copy of a data item to be locked

Each site maintains its own lock and it is allowed to grant or to reject a lock

If a transaction requesting a lock is granted the lock by majority of sites then it continues its execution; otherwise it fails and aborts

A transaction granted a lock informs all copies that it has been granted the lock

Voting creates a higher message traffic between the sites than distinguished copy technique



01 Transaction processing in DDBS

## References

Elmasri R., Navathe S. B., *Database Systems*, chapters 26.6, 26.7

`http://www.uow.edu.au/~jrg/235/HOMEWORK/`

11 How to process distributed database systems ?