REPLICATION PART 2

METHODS & CONTROLS

QUERY PROCESSOR

Query Processor (QP)

- **Handling SQL sentences**
  - **Parsing:** Checks syntax of SQL sentence and creates parse tree
  - **Validation:** Checks validity of SQL sentence
  - **Optimization:** Creates Execution Plan for the optimized cost

Prepare

Query Processor

- Execution: Run Execution Plan

Execute

Storage Manger

Execution Plan

- Path direction, order, methods are expressed in a tree format

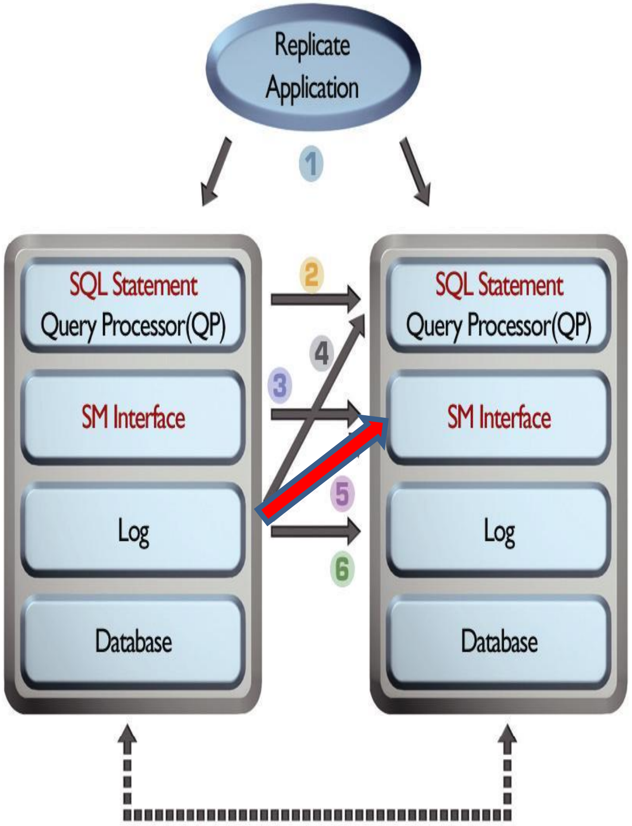
Storage Manager / SM

- Manages and saves data by concurrency control recovery

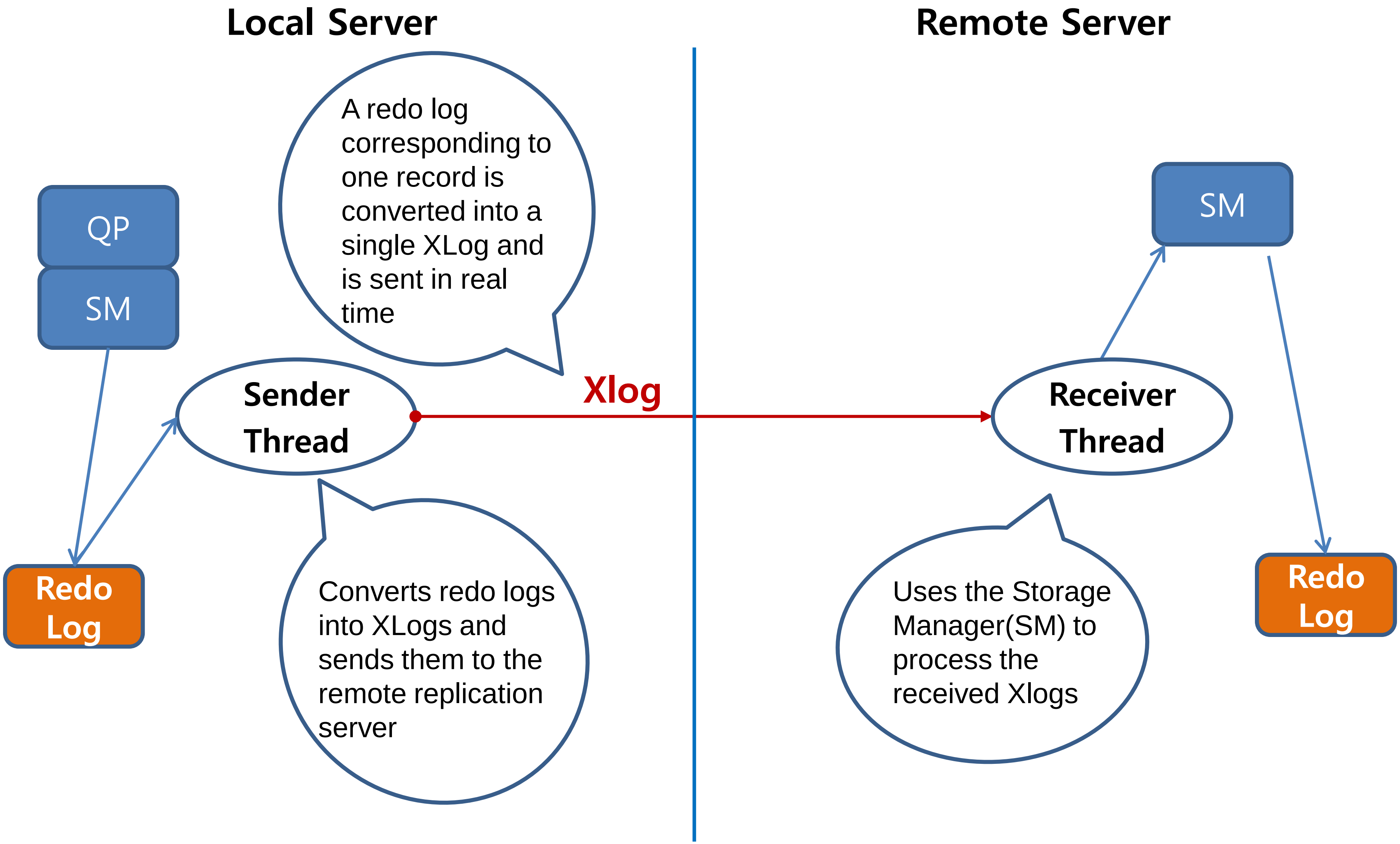
ALTIBASE HDB REPLICATION METHODS

ALTIBASE HDB Replication Methods

1. Handled by Client Application
   - Additional effort required to author applications and ensure data consistency
2. By Sending SQL Statements
   - Increased QP load and requirement to check for replication conflicts
3. By Sending SQL Statement Execution Plans
   - Increased network load due to high volume of transmission
4. By Sending Redo Logs and Converting Them to SQL
   - Increased SQL conversion expense and QP load
5. By Converting Redo Logs in the SM to a replayable Logical Form and Sending Them
   - Some conversion expense is incurred, but replication performance is good
6. By Sending Redo Logs and directly applying Them
   - Performance is good, but this method is not compatible with an Active-Active configuration



ALTIBASE HDB REPLICATION ARCHITECTURE



A redo log corresponding to one record is converted into a single XLog and is sent in real time

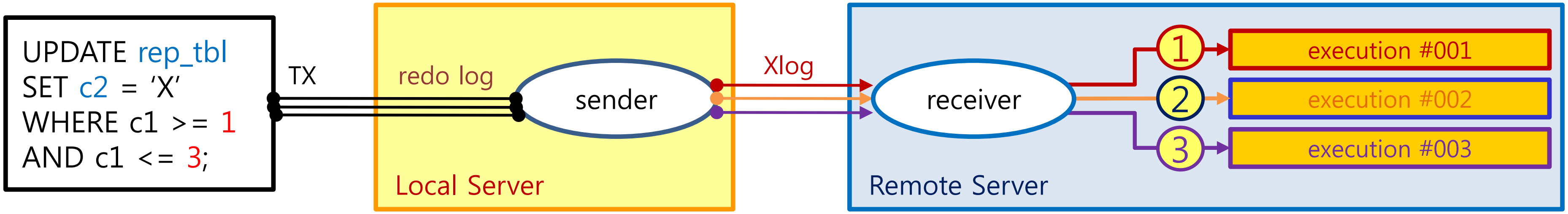
Converts redo logs into XLogs and sends them to the remote replication server

Uses the Storage Manager(SM) to process the received Xlogs

ALTIBASE HDB REPLICATION ARCHITECTURE

A Detailed Look at the SQL Statements Executed in the Course of Replication

- An UPDATE statement that changes the contents of 3 records is executed.
- The related redo logs are converted into 3 XLogs, which are then sent.
  - Individual XLogs are sent in real time as soon as they are created.
- A total of 3 records change operations, each corresponding to an individual XLog is executed on the remote server.

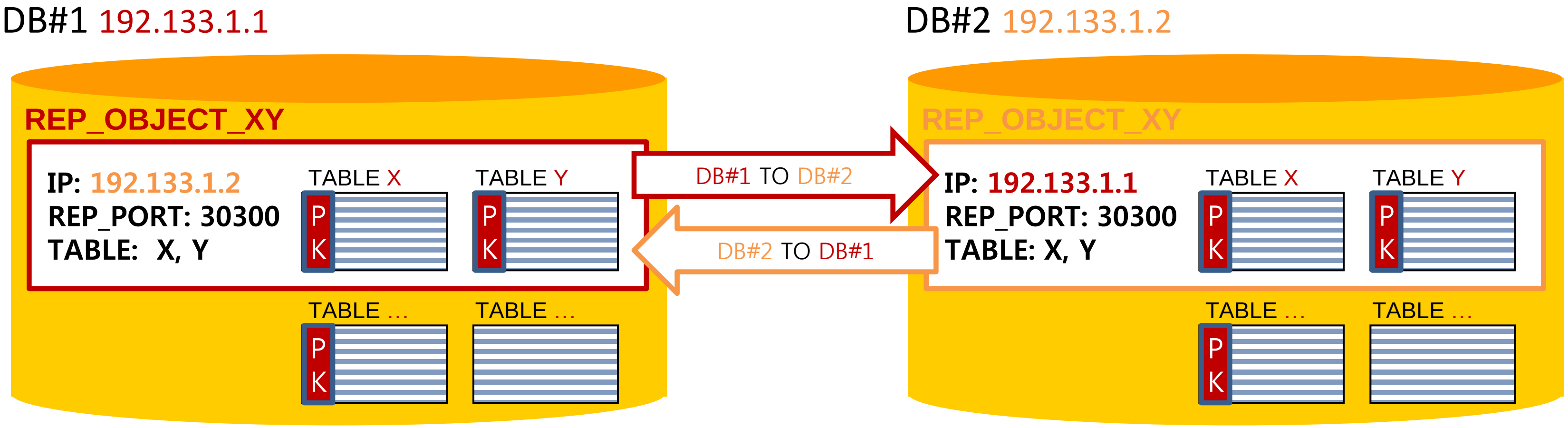


```
UPDATE rep_tbl
SET c2 = 'X'
WHERE c1 >= 1
AND c1 <= 3;
```

REPLICATION OBJECT

Replication Object

- An object that must exist on both the local server and the remote server in order for replication to be possible
  - Name of replication objects have to be the same among servers
  - Replication objects cannot be reassigned to other replication objects.(Only 1:1)
- Identified by IP and replication port number
  - Up to 32 replication objects can be created in a single database.
- Multiple IPs per object are supported.
- Each pair of replication objects is set to either LAZY or EAGER mode.



REPLICATION OBJECT

Condition for replication tables

- Must have PK
- The name and specifications for replication columns have to be identical
  - The column to be replicated is identified by its name out of all the records
  - The number of columns and its orders are unnecessary when they are different from local to remote server
  - When data exists in a local server but not in a remote server, the NULL is automatically assigned in a remote server
  - The error is occurred when the specifications are different such as data type, length, constraints when column name is the same
  - FK is not available

Misunderstanding about replication

- Is Sequence available in replication?
  - As the replication is table based, the objects that excludes table are not allowed such as sequence

REPLICATION OBJECT

Creating replication object

```
CREATE [LAZY|EAGER] REPLICATION replication_name [AS MASTER|AS SLAVE]
[OPTIONS options … [options … ] ]
WITH {'remote_host_ip ' , remote_replication_port_no }
FROM user_name.table_name TO user_name.table_name ,
[FROM user_name.table_name TO user_name.table_name ] ;
```

- **LAZY, EAGER :** Replication type, assigned as LAZY when it's omitted
- **MASTER, SLAVE :** Assigns server's role for Conflict Resolution. Not assigned when it's omitted
- ***option*** **:** Additional functions for replication object such as offline replication
- ***replication_name*** **:** Replication object name and they have to be identical between servers
- ***remote_host_ip*** **:** IP address of remote server
- ***remote_replication_port_no*** **:** Replication receive port number of remote server
- **FROM - TO :** Target replication table has to be assigned as " From Local – TO Remote"

REPLICATION OBJECT

Example of Creating Replication Object

- **Local Server (IP: 192.168.1.160, Replication Port:25524)**

```
CREATE REPLICATION rep1
WITH '192.168.1.12',35524
FROM sys.employee TO sys.employee,
FROM sys.department TO sys.department;
```

- **Remote Server (IP: 192.168.1.12, Replication Port:35524)**

```
CREATE REPLICATION rep1
WITH '192.168.1.160',25524
FROM sys.employee TO sys.employee,
FROM sys.department TO sys.department;
```

REPLICATION OBJECT

Add/Delete/Configure replication server IP(Multiple IP config.)

```
ALTER REPLICATION replication_name  {ADD|DROP|SET} HOST
‘remote_host_ip’ , remote_replication_port_no ;
```

Add/Delete replication table

```
ALTER REPLICATION replication_name {ADD|DROP} TABLE
FROM user_name.table_name TO user_name.table_name ;
```

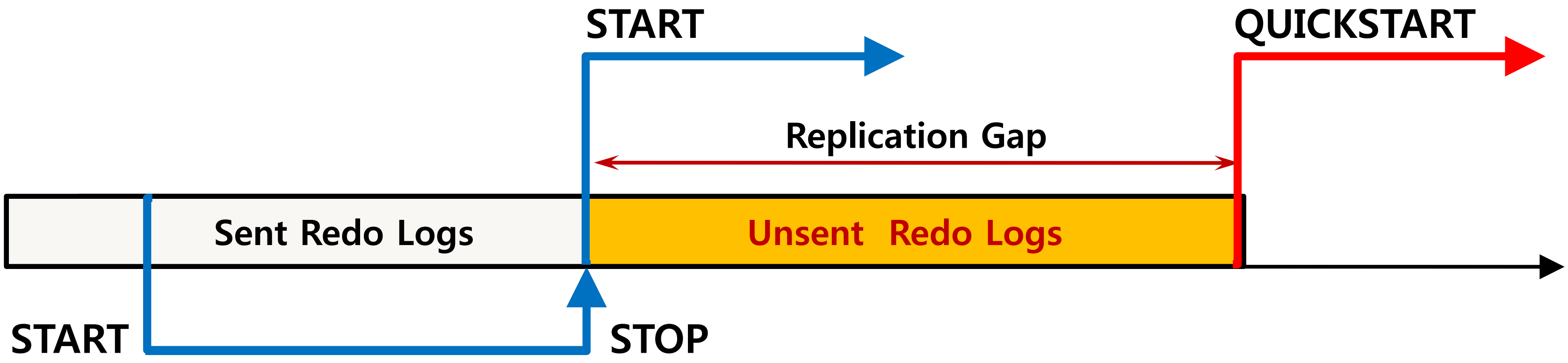
Delete replication object

```
DROP REPLICATION replication_name ;
```

REPLICATION CONTROL

Replication Control

- START – Starts replication from the last replication point(Normal Start)
- QUICKSTART – Skips all the ‘Unsent Redo Logs’ and starts from most recent point (For a emergency)
- STOP – Stops replication at current time



Syntax

```
ALTER REPLICATION replication_name {START|QUICKSTART|STOP} ;
```

REPLICATION TABLE CLONE

Table Clone

- Cloning table is available using replication object
- Inserting all the records from local table to remote table
  - Only available when replication is stopped
  - Replication starts automatically once cloning is completed(START)
- Used when recovering particular table or node and when adding a new node

Simple Procedure

- 'TRUNCATE' the target table
- Execute 'table clone statement' at the local server

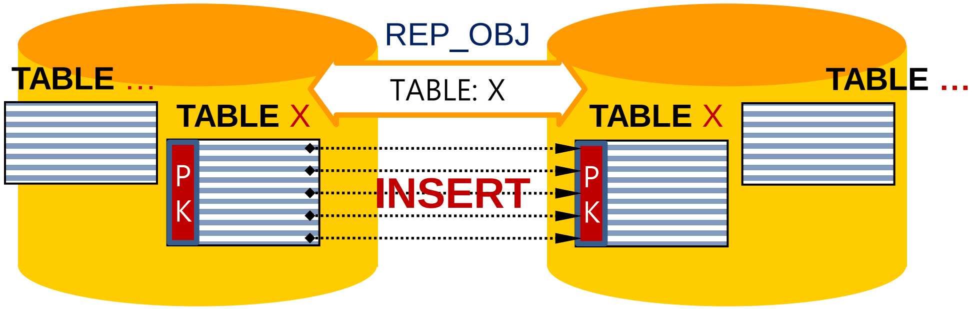


TABLE CLONE

Syntax

```
ALTER REPLICATION replication_name SYNC [ONLY]
[PARALLEL parallel_factor ]
[TABLE user_name.table_name, … , user_name.table_name ];
```

- **SYNC:** Automatically starts replication after table cloning is executed
- **ONLY:** Only table cloning is executed but it does not start (START) replication automatically
- ***parallel_factor* :** Number of threads that will execute table cloning(Default 1, Maximum CPU*2)
- **TABLE:** Assign a target table for cloning but entire tables are applied when this option is omitted

TABLE CLONE

Cautions

- It shouldn’t be executed on a Active-Active environment when a system is operating
  - Other tables of replication object are not replicated until the table cloning is completed
- It has to be executed after all the table records from remote servers are deleted
  - Cloning fails due to INSERT conflicts occurred when the same PK exists
  - There may be a overloading on DBMS due to INSERT operation of same PK
  - Large amount of INSERT conflicts errors recorded in replication trace log file
- ‘TRUNCATE’ statement is recommended when deleting table records from remote server
  - User’s mistake – When the records from local server is deleted as the records from remote server is deleted by ‘DELETE’ statement

Thank you!

Altibase Education Center

Tel : 02-2082-1451
Fax : 02-2082-1459
E-mail : education@altibase.com
Homepage : http://edu.altibase.com