

# Replication Part 5

## Optimization & Monitoring

# Optimization

# OVERVIEW

---

## ❖ Consideration for optimizing ALTIBASE HDB Replication

- ◆ Compulsory for minimizing the Replication gap and improving TPS
  - Optimize Network
  - Transaction Tuning
  
- ◆ Active-Active
  - Partial failure and partial roll back
  - Lock system in Replication
  - Replication lock timeout
  - Replication Dead-Lock
  - Replication Sender Tuning

# APPLICATION RESTRICTION

---

- ❖ **UPDATE for PK is not available**

- ◆ Treated as error when **UPDATE** is executed on PK as there could be a replication conflict

# NETWORK OPTIMIZATION

---

## ❖ H/W

- ◆ Connected with H/W performance as it's TCP/IP Network based replication
  - Replication exclusive line that is over 1 Giga Bit bandwidth separated from service line
  - Exclusive LAN card is recommended as much as the number of replication objects
  - Twice the number replication related network H/W is required to prepare for network failure(multiple IP)
  - Optimize other network facilities

## ❖ S/W

- ◆ Optimize the configuration of network parameters depending on OS platform

## ❖ Design Replication Object

- ◆ Separating replication object is recommended if the order of transaction among replication objects is not important
  - Increase of performance can be expected as the replication sender thread is created as much as the number of objects
  - Separation of disk table and memory table is recommended
    - Prevents the performance reduction of memory table reduced by the slow performance of disk table

# TRANSACTION TUNING

---

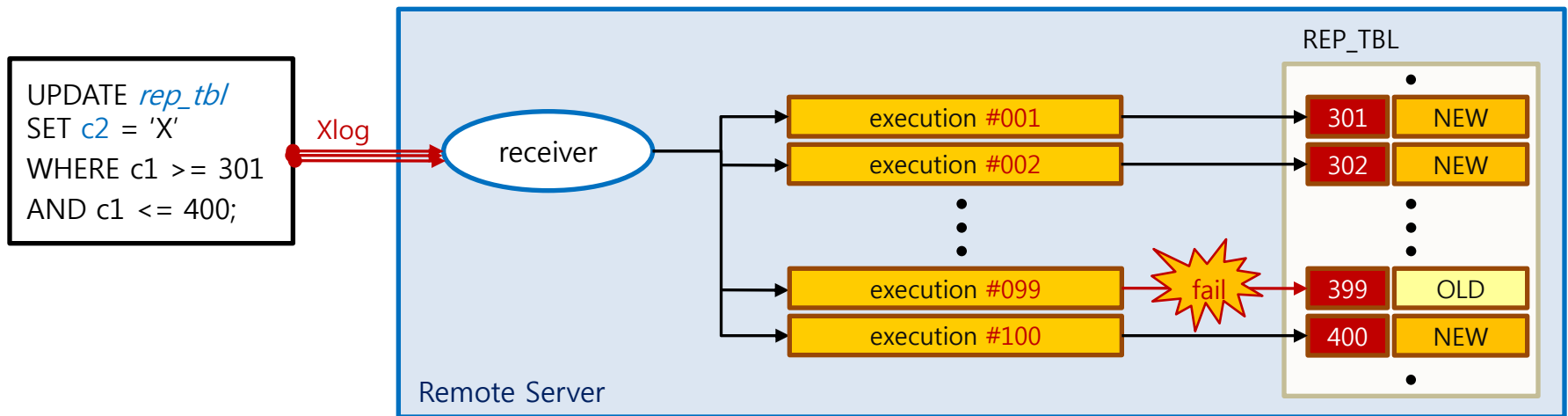
## ❖ Transaction tuning that exceeds network bandwidth

- ◆ Bulk DML operations are not recommended
  - Separation using LIMIT query
  - Execute same operation after configuring replication session control option to “NONE” from both servers
  - Use TRUNCATE rather than DELETE when deleting entire table regularly

# PARTIAL FAIL & PARTIAL ROLLBACK

## ❖ Partial Fail

- ◆ Applying particular record of transaction is failed from remote server
- ◆ Entire transaction fails as particular record fails to reflect



[ Situation where one transaction that modifies 100 records fails to reflect one data during replication ]

## ❖ Cause

- ◆ Replication conflicts and exceeding replication lock timeout

# PARTIAL FAIL & PARTIAL ROLLBACK

## ❖ Different Processing with different replication types for partial fail

	lazy	eager
Local	-	Error is occurred when COMMIT is executed
Remote	Error log is recorded with the record and its statement Execute partial rollback for the record and its operation	Error log is recorded about the record and its statement
Outcome	Particular record is only applied in local server	Transaction is not applied to both local and remote server

## ❖ Partial Rollback

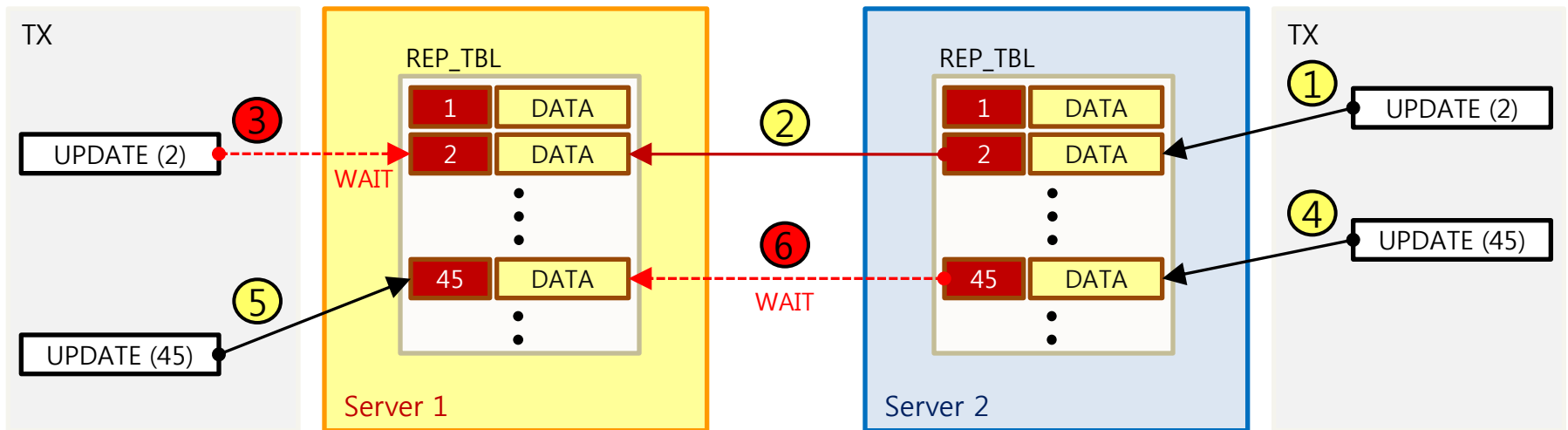
- ◆ Method that rollbacks the record that have failed during replication in remote server
- ◆ Purpose is to prevent the failure of reflection of entire transaction due to the failure of particular record
- ◆ Record where partial rollback is executed has the data inconsistency between local and remote server



# LOCK SYSTEM IN REPLICATION

## ❖ Lock System in replication

- It has the record unit lock system similar with single server
- ◆ Lock is acquired in two ways when DML is executed on the same records
  - When DML operation of replication acquired the lock first (1~3)
  - When DML operation of master transaction acquired the lock first(4~6)



[ Flow of acquiring lock between replication servers depending on its lock acquire point ]

# LOCK SYSTEM IN REPLICATION

---

## ❖ Situations for the lock types when long-lasting

- ◆ When DML operation of replication acquired the lock first
  - The master transaction is waiting for the lock to be acquired but it is normal flow in a DBMS level
- ◆ When DML operation of master acquired the lock first
  - Replication is stopped as the receiver waits to acquire the lock
  - Data inconsistency becomes more serious as replication gap increases

## ❖ How to handle

- ◆ Configure the waiting time limit for DML operation of replication
  - Choose the policy that gives up the DML operation of particular record when the waiting time is exceeded

# REPLICATION LOCK TIMEOUT

---

## ❖ Replication lock timeout

- ◆ Maximum waiting time for a receiver to acquire a lock during replication
- ◆ Purpose is to prevent the entire replication operations got affected by lock
- ◆ Applied by the property from replication receiver
  - REPLICATION\_LOCK\_TIMEOUT, 5 seconds by default

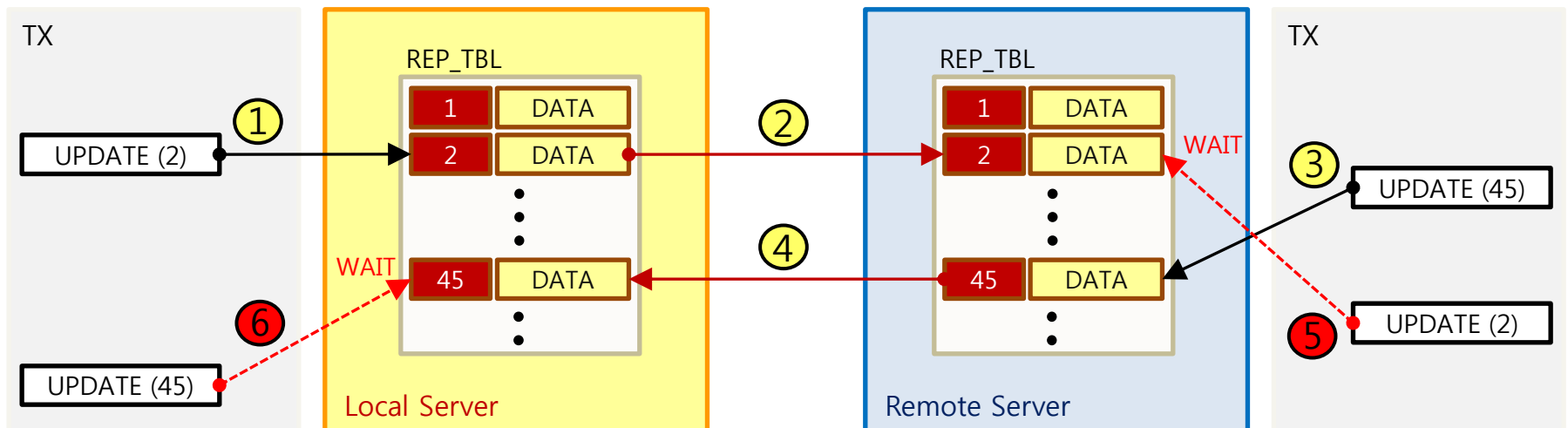
## ❖ Considerations

- ◆ Configure the appropriate replication lock timeout number depending on the system
  - If the number is too big, the replication might be stopped for a long time
  - If the number is too small, the data inconsistency might become more serious as the frequent lock timeout
- ◆ DML execution from replication server
  - Caution is advised that some records might not be applied due to replication lock timeout

# REPLICATION DEADLOCK

## ❖ Replication Deadlock

- ◆ Infinite waiting as the DML transaction of master and a DML operation of Replication are crossed each other
- ◆ Network deadlock that cannot be detected from DBMS
- ◆ It can be controlled by configuring the maximum waiting time of DML transaction from the property
  - UTRANS\_TIMEOUT
- ◆ Best way is to implement the system that considers two DML transactions compete



# REPLICATION SENDER TUNING

## ❖ Sender's approaching order of redo log

1. Replication redo log buffer
2. Redo log buffer
3. Redo logfile

## ❖ Tuning via property when sender's performance decreased

- ◆ Increment the size of replication redo log buffer size

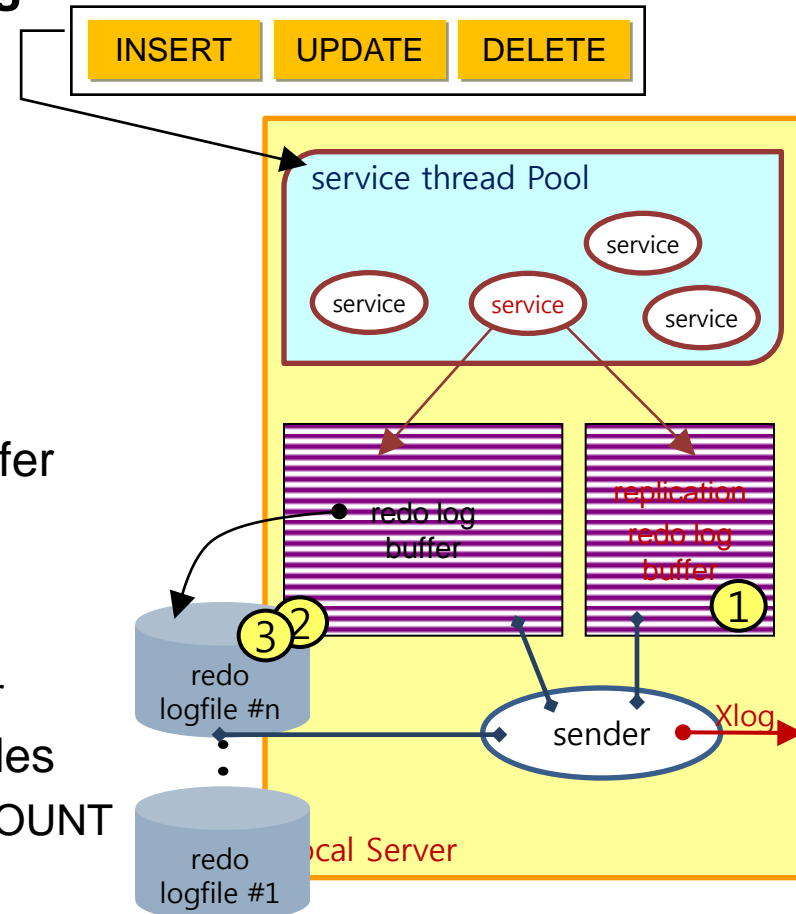
REPLICATION\_LOG\_BUFFER\_SIZE

- Maximum 4G and 30M by default
- To not directly approach the redo log buffer

- ◆ Increment the number of prepared redo logfiles

REPLICATION\_PREFETCH\_LOGFILE\_COUNT

- Maximum 1024, 0 by default
- Performance is improved as redo log files are read in advance
- 2<sup>nd</sup> tuning target



# Monitoring

# OVERVIEW

---

## ❖ DBMS Monitoring methods

Methods	Description
Internal Monitoring	Monitors internal DBMS by inquiring the meta tables and performance views
External Monitoring	Monitors external categories that are related to DBMS using OS commands in a OS level
Trace Log Monitoring	Monitors various trace log messages created from DBMS

## ❖ How to do Monitoring

- ❖ Create a shell script that include all the related commands and run on regular basis
  - Use Utility
    - ALTIMON
    - Replication Manager
- ❖ Write an application program

# REPLICATION MONITORING

---

## ❖ Main internal monitoring categories for replication

- ◆ Replication Gap
- ◆ Sender
- ◆ Receiver

## ❖ Main external monitoring categories for replication

- ◆ Network
- ◆ Redo log file system
- ◆ ALTIBASE HDB startup status
- ◆ OS startup status

## ❖ Main trace log categories for replication

- ◆ Replication trace log file



# REPLICATION MONITORING

---

## ❖ Additional categories for internal monitoring

- ◆ The following tables are essential as they are somehow effected each other even though they are not directly related
  - Executing BULK DML
    - v\$transaction, v\$statement, v\$session

# REPLICATION META TABLE

## ❖ Meta table related to Replication

Meta table	Description
<b>SYS_REPLICATIONS_</b>	<b>Replication object information</b>
<b>SYS_REPL_HOSTS_</b>	<b>IP information about target replication of each replication object</b>
<b>SYS_REPL_ITEMS_</b>	<b>Table information about target replication for each replication object</b>
<b>SYS_REPL_OFFLINE_DIR_</b>	Information about offline Replication option(5.3.3 higher)
<b>SYS_REPL_OLD_COLUMNS_</b>	Information about column of target Replication that sender is using
<b>SYS_REPL_OLD_INDEX_COLUMNS_</b>	Information about index column of target Replication that sender is using
<b>SYS_REPL_OLD_INDICES_</b>	Information about index of target Replication that sender is using
<b>SYS_REPL_OLD_ITEMS_</b>	Information about table of target Replication that sender is using
<b>SYS_REPL_RECOVERY_INFOS_</b>	Meta table of log information for Replication recovery

# REPLICATION META TABLE

## ❖ SYS\_REPLICATIONS

Column Name	Description
REPLICATION_NAME	Replication name
IS_STARTED	Whether replication is started or not
XSN	The restart SN that the sender will send Xlog
ITEM_COUNT	Number of replication tables
CONFLICT_RESOLUTION	Method of conflict resolution
REPL_MODE	Default Replication Mode

## ❖ SYS\_REPL\_HOSTS\_

Column Name	Description
HOST_NO	Host Identifier
REPLICATION_NAME	Replication Name
HOST_IP	Remote Server IP
PORT_NO	Remote Server Replication Port Number

# REPLICATION META TABLE

---

## ❖ **SYS\_REPL\_ITEMS\_**

Column Name	Description
REPLICATION_NAME	Replication name
TABLE_OID	Table Object Identifier
LOCAL_USER_NAME	Name of table owner from local server
LOCAL_TABLE_NAME	Name of local server's table
REMOTE_USER_NAME	Name of table owner from remote server
REMOTE_TABLE_NAME	Name of remote server's table

# REPLICATION PERFORMANCE VIEW

## ❖ Replication Performance View

- Available only when sender and receiver are activated
  - The replication gap number from v\$repgap table is shown only when sender is activated

Performance View	Description
v\$repgap	<b>Replication Gap Information</b>
v\$repsender	<b>Sender Information</b>
v\$repsender_transtbl	Transaction table information of Sender
v\$repreceiver	<b>Receiver Information</b>
v\$repreceiver_column	The information of replication column from Receiver
v\$repreceiver_transtbl	The information of transaction table from Receiver
v\$repsync	The information of table that is performing table clone
v\$repoffline_status	The information of offline replication state (5.3.3 higher)
v\$repexec	The information of replication owner
v\$repllogbuffer	The information of replication log buffer

# REPLICATION META TABLE

## ❖ v\$repgap

Column Name	Description
REP_NAME	Replication Object Name
REP_LAST_SN	Serial number of last log record
REP_SN	Serial number of log record that has currently been sent
REP_GAP	The difference between REP_LAST_SN and REP_SN
READ_FILE_NO	The log file number of that the DB is currently reading

## ❖ v\$repsender

Column Name	Description
REP_NAME	Replication Object Name
XSN	SN of log record that has currently been sent
COMMIT_XSN	SN of current log record
STATUS	Current Status
SENDER_IP	IP address of sender
PEER_IP	IpAddress of remote
REPL_MODE	Replication mode that the user assigned

# REPLICATION META TABLE

---

## ❖ v\$repreceiver

Column Name	Description
REP_NAME	Replication Object Name
MY_IP	IP address of local server
PEER_IP	IP address of remote server
APPLY_XSN	Handling XSN

# REPLICATION MONITORING EXAMPLE

## ❖ Local Server

```
iSQL> SELECT rep_name, rep_sn, rep_last_sn, rep_gap, read_file_no FROM v$repgap;
```

REP_NAME	REP_SN	REP_LAST_SN	REP_GAP	READ_FILE_NO
REP1	60602217	61617892	1015675	529

```
1 row selected.
```

```
iSQL> SELECT rep_name, xsn, status, repl_mode FROM v$repsender;
```

REP_NAME	XSN	STATUS	REPL_MODE
REP1	60602217	1	LAZY

```
1 row selected.
```

```
iSQL> SELECT replication_name, xsn, is_started FROM SYSTEM_.SYS_REPLICATIONS_;
```

REPLICATION_NAME	XSN	IS_STARTED
REP1	60600750	1

```
1 row selected.
```

```
iSQL> exit
```

```
$
```

```
$ ls $ALTIBASE_HOME/logs
```

```
loganchor0 logfile529 logfile532 logfile535 logfile538 logfile541 logfile544
loganchor1 logfile530 logfile533 logfile536 logfile539 logfile542 logfile545
loganchor2 logfile531 logfile534 logfile537 logfile540 logfile543
```



# REPLICATION MONITORING EXAMPLE

---

## ❖ Remote Server

```
iSQL> SELECT rep_name, apply_xsn FROM v$repreceiver;
```

```
REP_NAME          APPLY_XSN
```

```
-----
```

```
REP1              60600750
```

# Q & A

# Thank you!

Altibase Education Center

Tel : 02-2082-1451

Fax : 02-2082-1459

E-mail : [education@altibase.com](mailto:education@altibase.com)

Homepage : <http://edu.altibase.com>