Altibase FAQ | Replication

Replication Overview

Background

Altibase's network-based replication architecture, while providing very speedy performance, imposes very little overhead on computing resources since it only transforms transaction logs into logical logs and sends them to remote servers for processing.

Basic Flow of Replication.

Unknown macro: 'gliffy'

Terminology

Technical term	Description
SN	Sequence Number of a transaction log (a unique number assigned to a transaction log)
XSN	Xlog Sequence Number (a unique number assigned to an Xlog)
xLog	Compact packet made by the sender to replay the same transaction in a remote server.
Sender	Thread that sends an information of a transaction from a local server to a remote server.
Receiver	Thread that receives xLog from the sender and applies it to the remote server.

Check

Query Processor (QP)	the internal module that deals with SQL execution. (not a thread)
Storage Manager (SM)	the internal module that reads record or writes transaction logs. (not a thread)

Description of Replication steps

Flow	Description
1,2	QP prepares SQL to execute. (Click here if you want to know more about how a SQL is executed in HDB.)
3	SM writes a log of the transaction.
4	Sender reads the log and converts it to xLog
5	Sender sends xLog to a receiver. If it is successful, sender's work is done.(Asynchronous Mode)

© Altibase Corp.

Flow	Description
6	Receiver tries to apply XLog to the remote server.
7	Receiver sends ACK to the sender after it applies xLog.

> **Check**
>
> Receiver directly applies Xlog to SM since XLog is already verified by QP of the local server. This method enables the best performance among all Replication techniques.

What is XLog?

Replication Manager has two threads, Sender and Receiver.
When the sender sees a new log, it immediately creates an XLog and sends it to the Receiver.
XLog contains the information about how to replay the same transaction in the remote server.

Types of XLog	Description
INSERT	Target Table id + Primary key values + data values
UPDATE	Target Table id + Primary key values + Values Before updated + Values After updated
DELETE	Target Table id + Primary key values

- INSERT transaction normally needs all information about the target table and the value of all columns.
- DELETE transaction needs the information about the target table and the primary key of the record to be deleted.
- UPDATE transaction needs the information about the target table, the previous value and the after value of the column. The previous value is necessary for the UPDATE transaction since the value at the remote server can be different from the value in the local server.(We will explain why the previous value in XLog exists in detail later.)

Types of Replication

Mode	Description
Asynchronous (LAZY)	Asynchronous mode does not check whether XLog is successfully applied to the remote server or not.
Synchronous (EAGER)	Synchronous mode always checks whether XLog is applied to the remote server without the conflict or not.

Unknown macro: 'gliffy'

In Asynchronous mode, a local transaction is completed with steps (1 + 2 + 3) - red box in above diagram. Replication transaction is completed with steps (4 + 5) - blue box in above diagram.
Steps (4 + 5) have no effect to the local transaction. In other words, the local transaction and the

© Altibase Corp.

replication transaction do not interfere each other.
This means that the local transaction can be completed even if the sender could not send Xlog to the remote server.
However, the local transaction in Synchronous mode is completed with all the steps (1 + 2 + 3 + 4 + 5 + 6 + 7).
The performance of Asynchronous mode is normally faster than Synchronous mode.

> **Check**
>
> After the Sender receives ACK from the receiver, the Sender tries to update "XSN" column of the meta table. In this way, the sender can determine which position of Xlog it has to send after Altibase is restarted.
> After the Receiver completes applying the XLog to Altibase, it updates "Apply XSN" column of the meta table. This way, the Receiver remembers up to which position it has to apply Xlog before Altibase stops.
> When the system is recovered from the failure condition, the Sender automatically sends Xlog from position it marked according to the column of the meta table indicates.

How to create Replication?

Replication can be initiated in few simple steps.
Step1. configure altibase.properties in $ALTIBASE_HOME/conf/altibase.properties.

```
REPLICATION_PORT_NO = 30300 (example)
```

Step2. Create the replication object and start it.

```
iSQL> CREATE REPLICATION rep1 WITH '192.168.1.30', 30300
FROM SYS.TABLE_1 TO SYS.TABLE_1,
FROM SYS.TABLE_2 TO SYS.TABLE 2,
...
;
iSQL> ALTER REPLICATION rep1 START;
iSQL> ALTER REPLICATION rep1 STOP;
iSQL> DROP REPLICATION rep1;
```

> **Check**
>
> You have to configure the property and specify the replication object in all nodes in the replication environment.

Please refer to Replication User's Manual for detailed information.

© Altibase Corp.

Summary

Altibase's log-based replication architecture, while providing very speedy performance, imposes very little overhead on computing resources since it only transforms transaction logs into logical logs and sends them to remote servers for processing.

Altibase replication provides users with a choice of two commonly used modes of replication: Asynchronous (Lazy) mode and Synchronous mode (Eager). Lazy mode focuses on high performance while Eager mode focuses on data integrity and consistency.

In Asynchronous mode, a local transaction is completed without waiting for the remote server to successfully commit the same transaction. In Synchronous mode, however, the local transaction is completed concurrently with the remote server – only if both local and remote servers confirm the successful committed operation. While asynchronous mode provides higher performance, allows for geographically distributed nodes, and multiple servers in the topology; there are cases where some records may become out-of-sync. When using asynchronous mode, some records can be different for a period of time due to network latency or failure, or maintenance windows during which replication may be stopped by an administrator. The window during which the records are out-of-sync may lead to replication conflicts

Do you want to learn more about replication conflicts ?

© Altibase Corp.