WHITE PAPER

Replicating Data into Sybase IQ with Replication Server®

SYBASE®

TABLE OF CONTENTS

1 1. Introduction

1 1.1 Purpose

1 1.2 Scope

1 1.3 Copyright

2 2. Replication Server and Sybase IQ Setup Quick Steps

3 3. Replication Server and Sybase IQ Setup Details

3 3.1 Add and Configure the Maintenance User

3 3.2 Replication Server Connection to Sybase IQ

3 3.3 Create Subscription

4 3.4 Resume the DSI Thread

4 3.5 Replication Server Configuration

4 4. Replication Notes and Observations

4 4.1 Parallel DSI Threads

5 4.2 Datatypes

5 4.3 Replication Speeds

1. INTRODUCTION

1.1 PURPOSE

This document highlights key issues and areas of concern regarding replicating data in a near real-time fashion into Sybase IQ using Replication Server® (RS). Most of these notes and ideas come from real-world experiences at various Sybase customers.

Some benchmark information is also included to highlight when this solution is viable. Please note that replicating data directly into Sybase IQ from RS has some limitations. For instance, it won't be as fast as replicating into Sybase Adaptive Server® Enterprise, Adaptive Server Anywhere, or most other OLTP engines.

Replicating data is nothing new to Sybase Adaptive Server Enterprise or Sybase Adaptive Server Anywhere. It is, however, new to Sybase IQ. This document outlines the simplest and easiest way to replicate data into Sybase IQ from Sybase Replication Server.

Please note that no other users will be able to modify data in a given table while Replication Server is modifying that same table. This is the current functionality of Sybase IQ called "table versioning."

Currently, Sybase IQ requires that to see the new "version" of the data that has been replicated into Sybase IQ, the user or application issue a commit. The querying application must issue a commit before a select in order to see the new data.

1.2 SCOPE

This document attempts to cover most of the intricacies of replicating data in as near a real-time manner as possible by Replication Server and Sybase IQ.

This document is a work in progress and most points contained in it come from real-world experience. Whether to follow this document is at the sole responsibility and discretion of the reader and implementation team.

1.3 COPYRIGHT

This document is the property of Sybase, Incorporated. It may not be copied in whole or in part, or be presented to other parties without the prior written consent of Sybase, Incorporated.

2. REPLICATION SERVER AND SYBASE IQ SETUP QUICK STEPS

1. Add and configure the maintenance user to Sybase IQ first

    grant connect to MAINT_USER identified by MAINT_PWD;

    grant membership in group rs_systabgroup to MAINT_USER;

    grant DBA to MAINT_USER;

    grant all to MAINT_USER;

2. In RS, use the create connection syntax instead of using RS_INIT to add the replicate database

    create connection to IQ.iqdb

    set error class to rs_sqlserver_error_class

    set function string class to rs_sqlserver_function_class

    set username to MAINT_USER

    set password to MAIN_PWD

    go

3. Create the subscriptions (assumes a replication definition exists)

    create subscription NEW_SUB_NAME

    for REPDEF_NAME

    with replicate at IQ.iqdb

    without materialization

    go

4. If the DSI thread is down, resume the DSI connection to Sybase IQ

    resume connection to IQ.iqdb

    go

5. Replication Server Configuration

    use RSSD

    go

    rs_configure dsi_max_xacts_in_group, '10'

    go

3. REPLICATION SERVER AND SYBASE IQ SETUP DETAILS

3.1 ADD AND CONFIGURE THE MAINTENANCE USER

Add the maintenance user to Sybase IQ first with *grant connect* syntax. This will add the maintenance user to the Sybase IQ environment so that Replication Server can log into Sybase IQ and apply the changed data.

Place the maintenance user in the *rs_systabgroup* in Sybase IQ with the *grant membership* syntax. (Without this step, the calls that Replication Server makes to the functions *rs_getlastcommit*, *rs_updatelastcommit*, etc. will error out because they cannot be found in the database owned by the maintenance user.) This syntax would be *grant membership in group rs_systabgroup* to *MAINTENANCE_USER*.

The stored procedures and tables necessary for replication are owned by the *rs_systabgroup*. For Replication Server to function properly without function string modification, the maintenance user must be in that group.

Grant the maintenance user *all* and *dba* privs so they can modify the data; you could do grant access table by table, but I have always granted *dba* and *all* privileges as overkill. This is done by issuing grant all to *MAINTENANCE_USER* and *grant dba* to *MAINTENANCE_USER*.

3.2 REPLICATION SERVER CONNECTION TO SYBASE IQ

In RS, use the create connection instead of using *RS_INIT* to add the replicate database. The following is an example of how to accomplish this:

```
create connection to IQ.iqdb
set error class to rs_sqlserver_error_class
set function string class to rs_sqlserver_function_class
set username to MAINT_USER
set password to MAIN_PWD
go
```

3.3 CREATE SUBSCRIPTION

At this point, it is assumed that the source system has been added to the replication environment either by Sybase Central, *rs_init*, or by hand. The assumption is that the rep agent and LTI interface are in place and ready to replicate data from the source system. This includes marking the table(s) for replication (sp_setreptable) and creating the replication definition(s).

A replication subscription can now be created to replicate data into Sybase IQ.

This can be done via the following syntax:

```
create subscription NEW_SUB_NAME
for REPDEF_NAME
with replicate at IQ.iqdb
without materialization
go
```

3.4 RESUME THE DSI THREAD

Should the DSI thread be down for some reason (DSI created to suspend the connection, error, etc.), the connection needs to be resumed.

This is done by running:

resume connection to IQ.iqdb

go

3.5 REPLICATION SERVER CONFIGURATION

To achieve better throughput when applying the OLTP transactions to Sybase IQ, it is recommended that the DSI batch up as many transactions as possible when submitting them to Sybase IQ.

The rule of thumb is that roughly 2,000 rows should be inserted into a table before a commit is issued. This allows Sybase IQ to be fully optimized for OLTP speed when it comes to storing the OLTP transactions in its structures.

use RSSD

go

rs_configure dsi_max_xacts_in_group, '10'

go

4. REPLICATION NOTES AND OBSERVATIONS

4.1 PARALLEL DSI THREADS

Parallel DSI threads cannot be used with Sybase IQ 12. This is in large part because of table versioning; there is no way in Replication Server to specify that certain tables will be handled by various DSI threads. Sybase IQ 12 cannot have multiple updates to the same table take place simultaneously.

As a side note, it would be possible to create multiple connections to Sybase IQ and have various tables apply transactions on these connections. This method is often referred to as *Parallel Connections* rather than the supported functionality of *Parallel DSI Threads*.

Parallel connections means that a second, third, etc., connection is created to the same Sybase IQ servername all with a different database name. Sybase IQ ignores the database name, so having multiple connections with different database names does not affect Sybase IQ and replication into Sybase IQ. Once the connections are created, subscriptions (tables) would be bound to a particular connection so that none of the connections would attempt to update tables in use by other connections.

The only drawback to this method is that transactions from one connection could show up in Sybase IQ before those on another connection. For instance, it is possible for newer transactions to be applied before the older transactions. This could happen if certain connections and queues are filling up while others are moving data through more quickly, which could cause certain reports and queries to return interesting results sets to the end user.

4.2 DATATYPES

The following datatypes raise concerns and issues when replicated into Sybase IQ:

- text
  - Replication Server handles text and image datatypes very differently than native datatypes
  - It is possible to replicate these types into Sybase IQ, but work with function strings will have to be done
- image
  - Replication Server handles textand image datatypes very differently than native datatypes
  - It is possible to replicate these types into Sybase IQ, but work with function strings will have to be done
- rawobject
  - Replication Server handles the rawobject datatype very differently than native datatypes
  - It is possible to replicate these types into Sybase IQ, but work with function strings will have to be done
- timestamp
  - must be replicated as binary(8)
  - TIMESTAMP in Sybase IQ is analogous to DATETIME in ASE
  - No ASE-like behavior of the "timestamp" column is in Sybase IQ

4.3 REPLICATION SPEEDS

The big question is: Should data be replicated into Sybase IQ straight from Replication Server? The answer depends on the data being replicated into Sybase IQ.

Sybase IQ 12.4.3 could sustain between 2,000 and 3,000 rows per hour.

Sybase IQ 12.5 could sustain between 3,000 and 16,000 rows per hour.

Sybase IQ 12.6 can sustain between 30,000 and 400,000 rows per hour!

The following table highlights the results from testing that was completed on a Sun® V440 with 4x1.2 GHz UltraSparc IIIi CPUs and 16 GB RAM. The system had 2x1 Gbit Fiber HBAs each connected to a Sun 3510 Storage Array.

The same number and type of devices and filesystems was used in the tests: 4 main store raw devices of 20 GB each and 4 temp store raw devices of 10 GB each. The system catalog was stored on a filesystem that was composed of a RAID 5 device spread across 12 disk drives. The same Sybase IQ configuration file (params.cfg) was used in all tests.

Columns Loaded	Table Width (bytes)	Sybase IQ 12.5 Rows Per Hour	Sybase IQ 12.6 Rows Per Hour
10	70	16,364	302,021
10	170	15,721	257,143
20	340	8,145	133,333
30	510	5,488	90,000
40	680	4,128	70,588
50	850	3,318	57,143

The columns were designed like this:

Columns Loaded	Column Specification
10	5 ints, 5 varchar (10)
10	5 ints, varchar (10/20/30/40/50)
20	10 ints, 2 x varchar (10/20/30/40/50)
30	15 ints, 3 x varchar (10/20/30/40/50)
40	20 ints, 4 x varchar (10/20/30/40/50)
50	25 ints, 5 x varchar (10/20/30/40/50)

The key reason for the performance range is the data width. Slower performance results as more columns and wider data are replicated. Very narrow tables (5-10 columns) have shown to be exceptionally fast, because the amount of work in Sybase IQ to modify those columns and indexes is less than in larger tables.

Sybase IQ is not an OLTP system; it is not designed to do single row inserts as fast as an OLTP engine like Sybase ASE. It is designed to load mass quantities of data, in bulk, quickly and efficiently.

The advent of Sybase IQ 12.6 has brought OLTP performance in Sybase IQ to an acceptable level when trading simplicity for performance. The ease of setting up a replication environment to replicate data directly to Sybase IQ is much more advantageous than having to build a custom solution to move data from the OLTP system into Sybase IQ.

SYBASE®